Hand On Lab 2   : Installing Docker CE on Ubuntu Server 18.04

Duration: 30 Minutes

Objectives:

To learn installation of docker & docker-compose on Ubuntu box.

Prerequisites:

1. Azure Subscription
2. Internet Connection
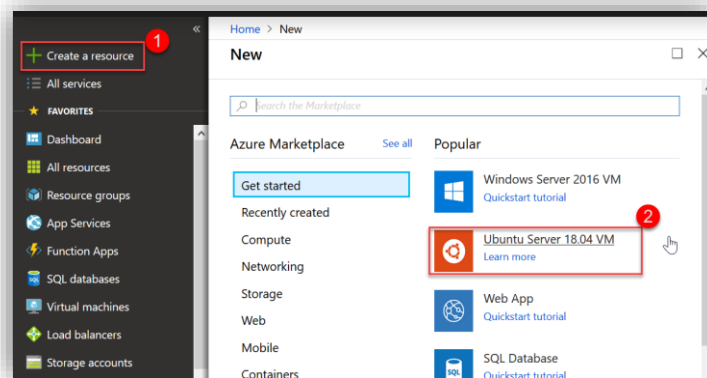3. SSH Client (PuTTY or Ubuntu Bash or Git Bash for Windows client)

   NOTE : MacOS and Linux have built in ssh client.

4. Web Browser (Any)

Steps :

I.   **Create Azure VM for Ubuntu Server 18.04 LTS**

1. Login into your azure portal (https://portal.azure.com)
2. Click on "Create Resource" to start creating new windows VM.
   Click on "Ubuntu Server 18.04"

3. Create new / Use existing resource group, enter name of VM as "DockerVM2" and then scroll down for more settings.

4. Choose "Password" for authentication and then enter user/password. Also need to open port 22 for SSH access.

Click on "Next: Disk >" for disk configuration.

5. Click "Next: Networking >" to accept the defaults and skip disk management.

6. Click "Next: Management >" to accept the defaults and skip networking.

7. Now, Turn Off boot diagnostic and click on "Review + Create"

8. Click "Create" on final validation page to start provisioning the VM.

9. Click on "Go to resource" once deployment finishes.
10. Click on connect to retrieve ssh connection parameters

II.   **Installation of Docker CE & docker-compose**
1.   Open Ubuntu bash (on Windows 10 pro) or Git Bash (on all windows)
2.   Enter ssh connection string copied from step 10.
     Then enter "yes" when prompted
     Enter password when prompted for password
     NOTE : Linux terminal should never echo password or any masked character (*).

3.   Try running "docker info" command to check if docker is pre-installed. You should get following message:

4.   As the message clearly suggests, use apt command to install docker.

NOTE : the additional "-y" is to supress prompt which confirms installation.
5.   Use command to enable current user to access docker cli without "sudo"

     $ sudo usermod -aG docker $USER

```
dockeradmin@DockerVM2:~$ sudo usermod -aG docker $USER
dockeradmin@DockerVM2:~$
```

6. Now, logout and login back [Linux Admins : reload current bash session]
Run command "exit" and then "ssh …" to reconnect

```
dockeradmin@DockerVM2:~$ sudo usermod -aG docker $USER
dockeradmin@DockerVM2:~$ exit      1
logout
Connection to 13.76.4.125 closed.
mahendra@Inspiron-13:~$ ssh dockeradmin@13.76.4.125    2
dockeradmin@13.76.4.125's password:
```

7. Test docker installation with command "docker info"

```
dockeradmin@DockerVM2:~$
dockeradmin@DockerVM2:~$ docker info
Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
Images: 0
Server Version: 17.12.1-ce
Storage Driver: overlay2
 Backing Filesystem: extfs
 Supports d_type: true
 Native Overlay Diff: false
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge host macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file logentries splunk s
Swarm: inactive
Runtimes: runc
```

8. Install "docker-compose" using following command:

   $ sudo apt install -y docker-compose

9. Few additional ports are required for next few HOLs, use portal to open them
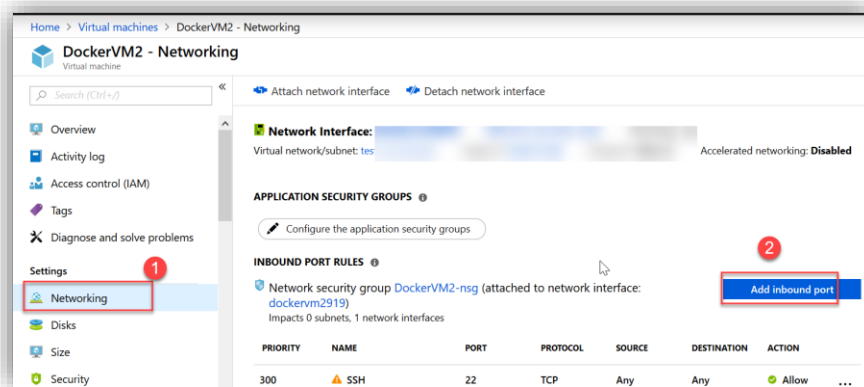
10. In "Source port ranges" type "*" and in target port ranges type "8000" then click "Apply"

11. Stop the VM from Azure portal