Containerization with Docker

New VM: docker01

IP Address: 10.0.5.12
Hostname: docker01-yourname

💡This may prove more interesting than first expected, since your docker system is running Ubuntu 20.04 cloud server.

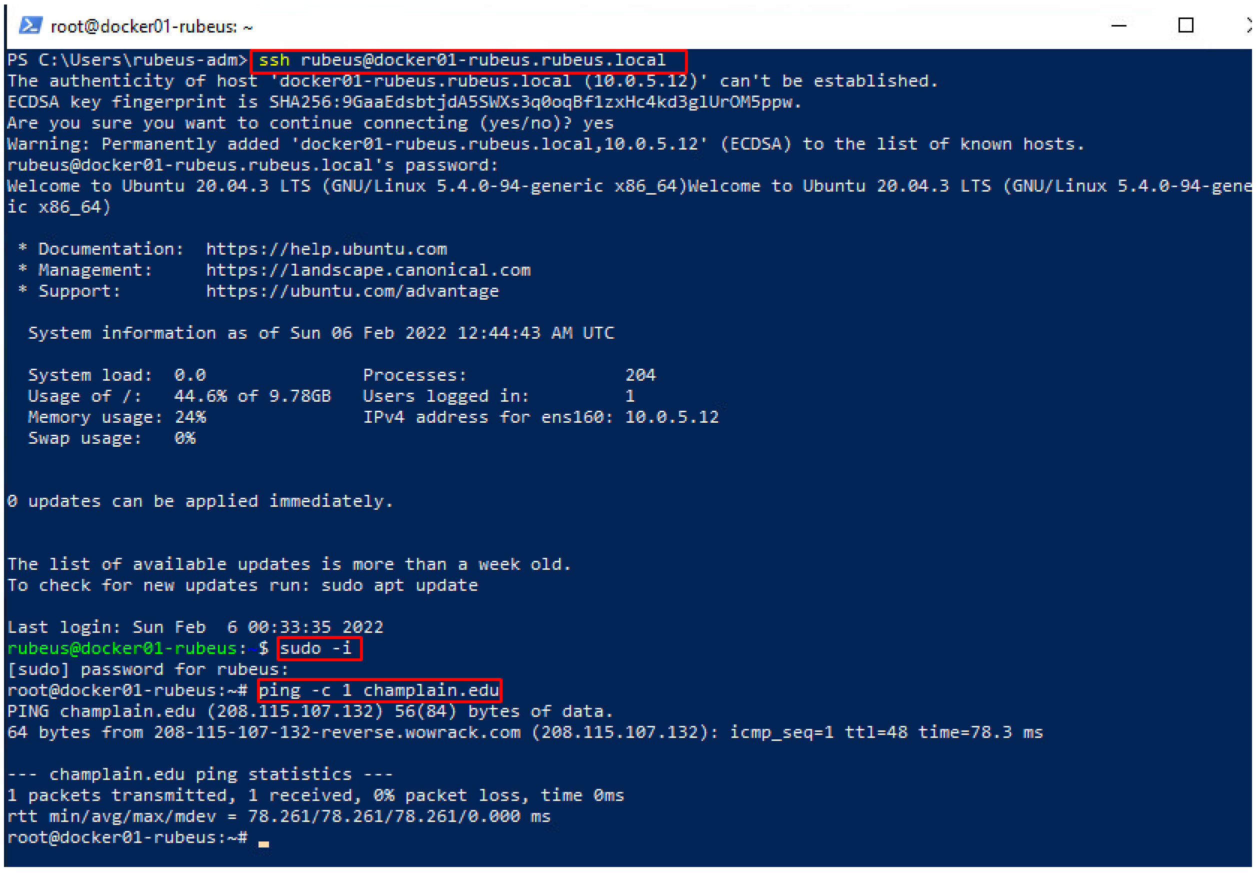
You will need to figure out how to use:

- netplan to configure a static IP address using /etc/neplan/00-installer-config.yaml
- update cloud.cfg to save the new hostname
- manually update the hostname
- the hosts file

Network system, DNS records, hostname, domain suffix, named sudo user, & disable remote root SSH … just via Ubuntu and not CentOS.

Note: Ubuntu has different groups for admins than CentOS. (Hint: use the id command as champuser to figure out what groups your named admin should be in)

```
Deliverable 1.  Screenshot showing PuTTY or powershell SSH session
from mgmt01 (use hostname, not ip address).  Elevate to root using
sudo -i and Within the session, ping champlain.edu.
```

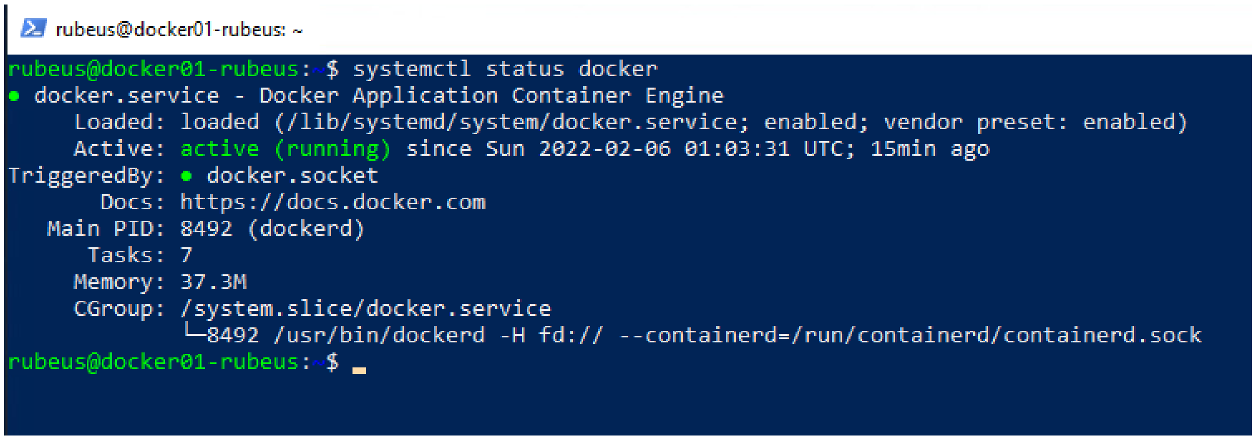


```
PS C:\Users\rubeus-adm> ssh rubeus@docker01-rubeus.rubeus.local
The authenticity of host 'docker01-rubeus.rubeus.local (10.0.5.12)' can't be established.
ECDSA key fingerprint is SHA256:9GaaEdsbtjdA5SWXs3q0oqBf1zxHc4kd3glUrOM5ppw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'docker01-rubeus.rubeus.local,10.0.5.12' (ECDSA) to the list of known hosts.
rubeus@docker01-rubeus.rubeus.local's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-94-generic x86_64)Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-94-gene
ic x86_64)

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

  System information as of Sun 06 Feb 2022 12:44:43 AM UTC

  System load:  0.0               Processes:               204
  Usage of /:   44.6% of 9.78GB   Users logged in:         1
  Memory usage: 24%               IPv4 address for ens160: 10.0.5.12
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sun Feb  6 00:33:35 2022
rubeus@docker01-rubeus:~$ sudo -i
[sudo] password for rubeus:
root@docker01-rubeus:~# ping -c 1 champlain.edu
PING champlain.edu (208.115.107.132) 56(84) bytes of data.
64 bytes from 208-115-107-132-reverse.wowrack.com (208.115.107.132): icmp_seq=1 ttl=48 time=78.3 ms

--- champlain.edu ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 78.261/78.261/78.261/0.000 ms
root@docker01-rubeus:~#
```

Install Docker

Follow the instructions for steps 1-3 on Digitalocean.com Community - How To Install and Use Docker on Ubuntu 20.04

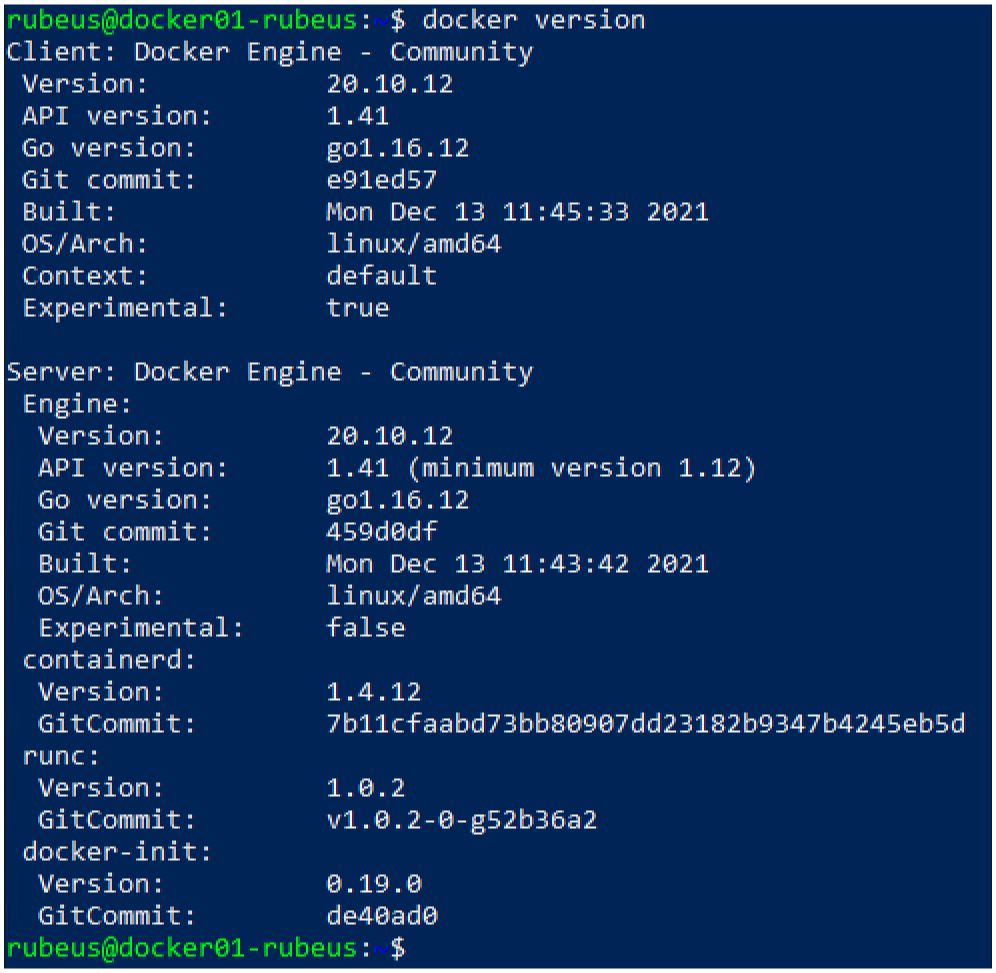
Deliverable 2. Confirm the Docker Service is running and provide a screenshot similar to the one below:



rubeus@docker01-rubeus: ~

```
rubeus@docker01-rubeus:~$ systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2022-02-06 01:03:31 UTC; 15min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 8492 (dockerd)
      Tasks: 7
     Memory: 37.3M
     CGroup: /system.slice/docker.service
             └─8492 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
rubeus@docker01-rubeus:~$ _
```

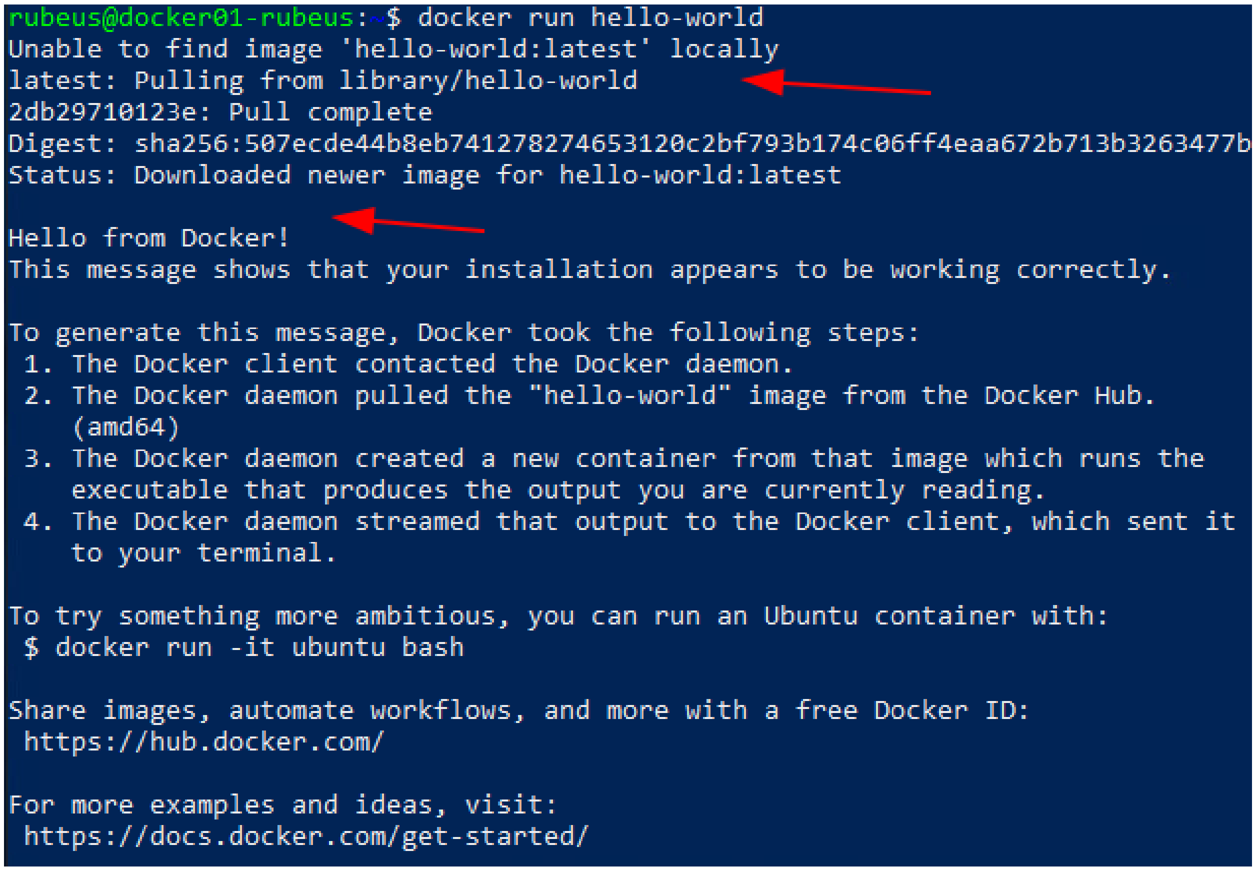
Deliverable 3. Confirm that your sudo user can access and print out version information using a screenshot similar to the one below



```
rubeus@docker01-rubeus:~$ docker version
Client: Docker Engine - Community
 Version:           20.10.12
 API version:       1.41
 Go version:        go1.16.12
 Git commit:        e91ed57
 Built:             Mon Dec 13 11:45:33 2021
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:          20.10.12
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.16.12
  Git commit:       459d0df
  Built:            Mon Dec 13 11:43:42 2021
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.4.12
  GitCommit:        7b11cfaabd73bb80907dd23182b9347b4245eb5d
 runc:
  Version:          1.0.2
  GitCommit:        v1.0.2-0-g52b36a2
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
rubeus@docker01-rubeus:~$
```

Docker Hello-World

`Deliverable 4. After running the docker hello world application as your named user & providing a screenshot similar to the one below, explain what has happened?`

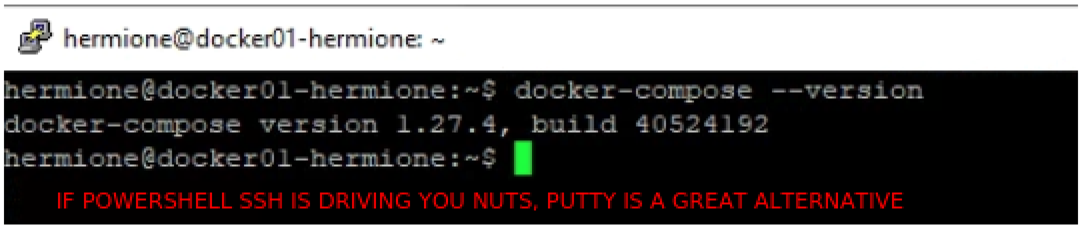


```
rubeus@docker01-rubeus:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:507ecde44b8eb741278274653120c2bf793b174c06ff4eaa672b713b3263477b
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Install Docker-Compose

Complete Step 1 in the following instructions to install docker-compose.

`Deliverable 5. Provide a screenshot similar to the one below that shows the docker-compose version.`



hermione@docker01-hermione: ~

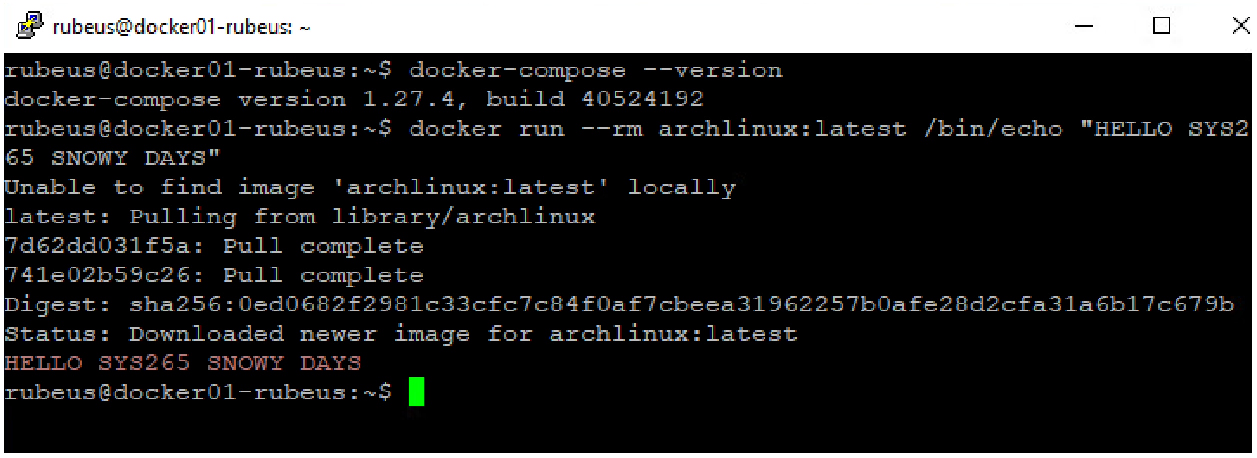
```
hermione@docker01-hermione:~$ docker-compose --version
docker-compose version 1.27.4, build 40524192
hermione@docker01-hermione:~$
```

IF POWERSHELL SSH IS DRIVING YOU NUTS, PUTTY IS A GREAT ALTERNATIVE

Hello SYS265

The following command pulls down an Arch Linux based docker image, invokes it in a container, and runs /bin/echo "HELLO SYS265 SNOWY DAYS " before deleting the container.

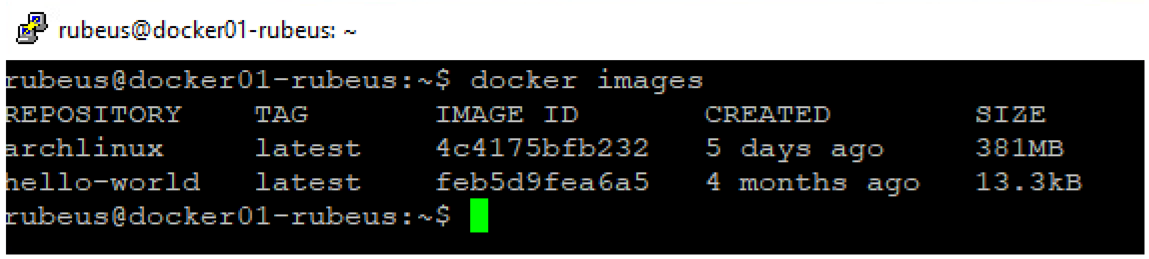
`Deliverable 6.  Provide a screenshot similar to the one below showing your "Hello Message":`



rubeus@docker01-rubeus: ~

```
rubeus@docker01-rubeus:~$ docker-compose --version
docker-compose version 1.27.4, build 40524192
rubeus@docker01-rubeus:~$ docker run --rm archlinux:latest /bin/echo "HELLO SYS2
65 SNOWY DAYS"
Unable to find image 'archlinux:latest' locally
latest: Pulling from library/archlinux
7d62dd031f5a: Pull complete
741e02b59c26: Pull complete
Digest: sha256:0ed0682f2981c33cfc7c84f0af7cbeea31962257b0afe28d2cfa31a6b17c679b
Status: Downloaded newer image for archlinux:latest
HELLO SYS265 SNOWY DAYS
rubeus@docker01-rubeus:~$
```

You can list the downloaded images on docker01 via the following command.



rubeus@docker01-rubeus: ~

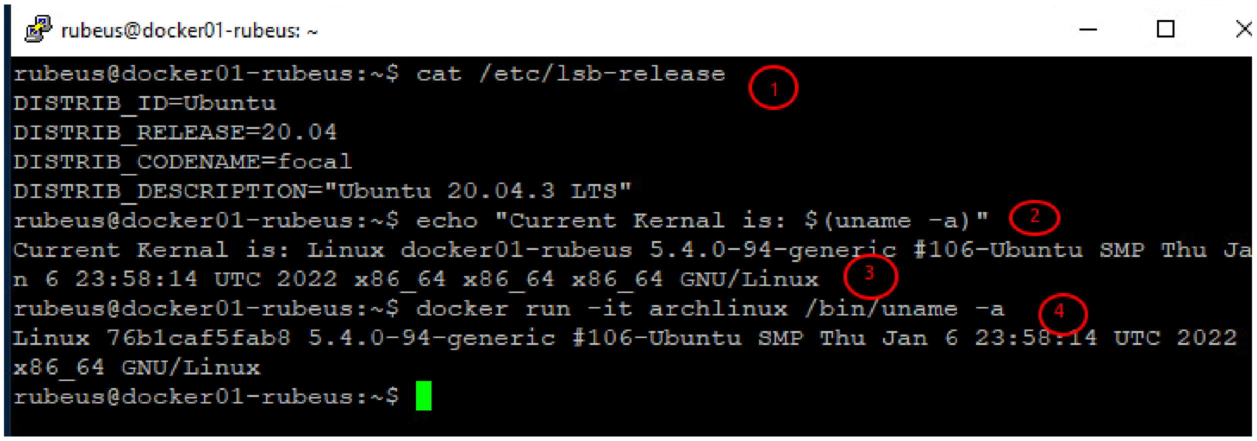
```
rubeus@docker01-rubeus:~$ docker images
REPOSITORY     TAG       IMAGE ID       CREATED         SIZE
archlinux      latest    4c4175bfb232   5 days ago      381MB
hello-world    latest    feb5d9fea6a5   4 months ago    13.3kB
rubeus@docker01-rubeus:~$
```

Docker Arch Linux Container

The following commands will:

1. Print out the current version of Ubuntu on docker01.
2. Print out the current version of docker01's linux kernel.
3. Invoke a container of the stored Ubuntu image as well as an interactive bash command prompt.
4. Print out the kernel being used by the Ubuntu container.

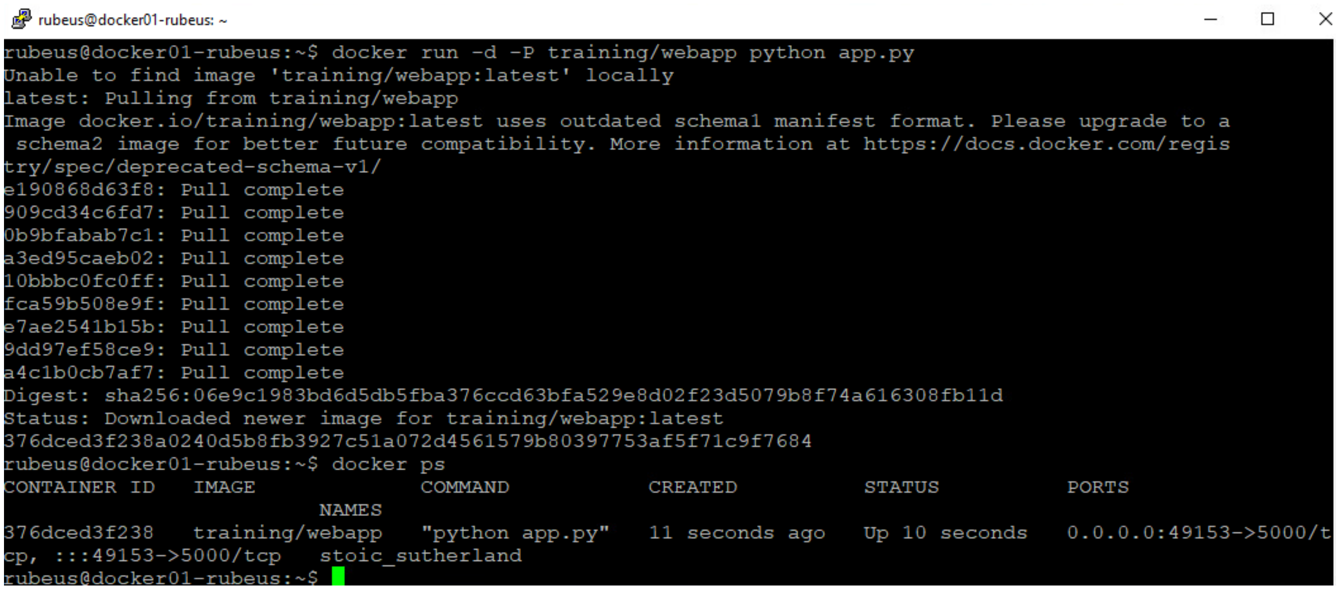
Deliverable 7. Provide a screenshot similar to the one below and an answer to the question: Based upon the version of kernels you see displayed within and outside of the container, what do you think is going on?



```
rubeus@docker01-rubeus:~$ cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.3 LTS"
rubeus@docker01-rubeus:~$ echo "Current Kernal is: $(uname -a)"
Current Kernal is: Linux docker01-rubeus 5.4.0-94-generic #106-Ubuntu SMP Thu Ja
n 6 23:58:14 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
rubeus@docker01-rubeus:~$ docker run -it archlinux /bin/uname -a
Linux 76b1caf5fab8 5.4.0-94-generic #106-Ubuntu SMP Thu Jan 6 23:58:14 UTC 2022
x86_64 GNU/Linux
rubeus@docker01-rubeus:~$
```

Docker Web Application

The following command will pull down the image, application and dependencies associated with a simple python web application.

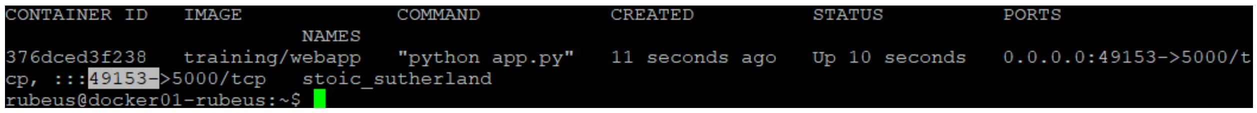


```
rubeus@docker01-rubeus:~$ docker run -d -P training/webapp python app.py
Unable to find image 'training/webapp:latest' locally
latest: Pulling from training/webapp
Image docker.io/training/webapp:latest uses outdated schema1 manifest format. Please upgrade to a
 schema2 image for better future compatibility. More information at https://docs.docker.com/regis
try/spec/deprecated-schema-v1/
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
10bbbc0fc0ff: Pull complete
fca59b508e9f: Pull complete
e7ae2541b15b: Pull complete
9dd97ef58ce9: Pull complete
a4c1b0cb7af7: Pull complete
Digest: sha256:06e9c1983bd6d5db5fba376ccd63bfa529e8d02f23d5079b8f74a616308fb11d
Status: Downloaded newer image for training/webapp:latest
376dced3f238a0240d5b8fb3927c51a072d4561579b80397753af5f71c9f7684
rubeus@docker01-rubeus:~$ docker ps
CONTAINER ID   IMAGE             COMMAND           CREATED          STATUS          PORTS
                        NAMES
376dced3f238   training/webapp   "python app.py"   11 seconds ago   Up 10 seconds   0.0.0.0:49153->5000/t
cp, :::49153->5000/tcp   stoic_sutherland
rubeus@docker01-rubeus:~$
```

`Deliverable 8.  Research the docker run command.  What does the -d and -P mean?`

Docker Networking

Take a look at your output, you should have a data element that looks similar to the one highlighted below, but likely not the same.

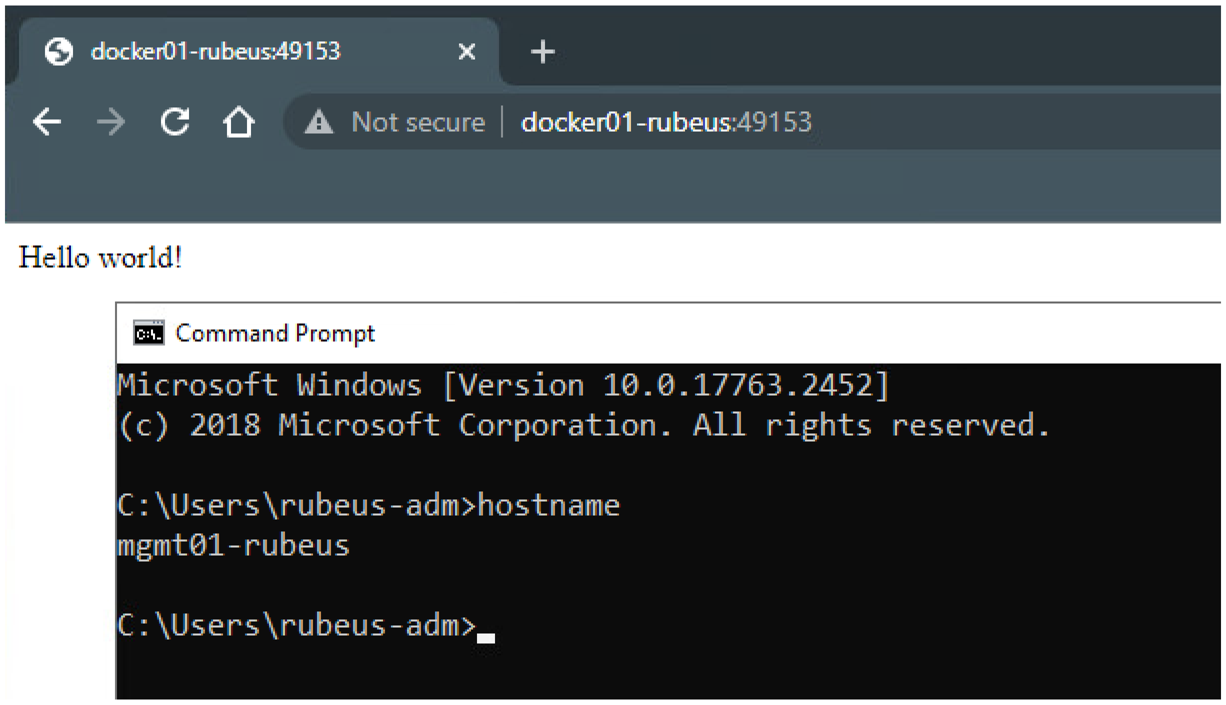


```
CONTAINER ID   IMAGE                COMMAND            CREATED          STATUS           PORTS
                    NAMES
376dced3f238   training/webapp    "python app.py"    11 seconds ago   Up 10 seconds    0.0.0.0:49153->5000/t
cp, :::49153->5000/tcp   stoic_sutherland
rubeus@docker01-rubeus:~$
```

We will call this “PortX” since we are rather creative.

Docker has configured **packet forwarding** on your base OS. In this case, traffic destined to host port PortX/tcp will be sent to the containerized application listening on 5000/tcp. You will need to allow the port (49153/tcp in this case) that shows up in docker ps through your firewalld firewall and reload.

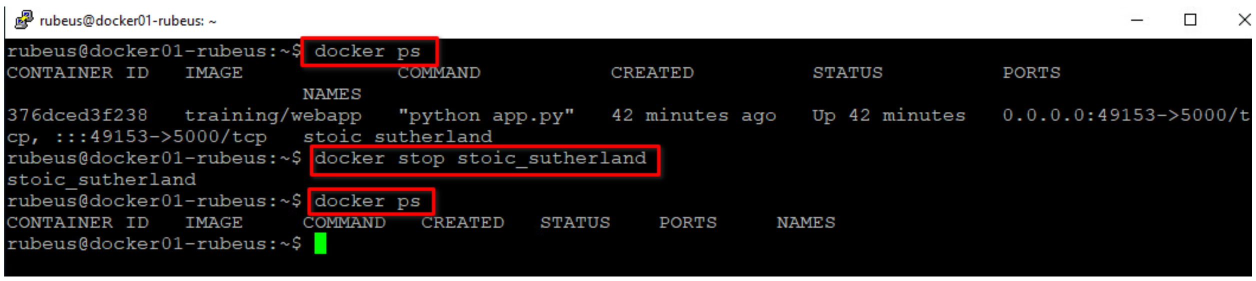
`Deliverable 9.  Screenshot showing a browsing session between mgmt01 and docker01 on the port shown in docker ps (you may have another port)`



Hello world!

```
Microsoft Windows [Version 10.0.17763.2452]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\rubeus-adm>hostname
mgmt01-rubeus

C:\Users\rubeus-adm>
```

Stop the testapp.



```
rubeus@docker01-rubeus:~$ docker ps
CONTAINER ID   IMAGE            COMMAND            CREATED          STATUS          PORTS
                        NAMES
376dced3f238   training/webapp   "python app.py"    42 minutes ago   Up 42 minutes   0.0.0.0:49153->5000/t
cp, :::49153->5000/tcp   stoic_sutherland
rubeus@docker01-rubeus:~$ docker stop stoic_sutherland
stoic_sutherland
rubeus@docker01-rubeus:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
rubeus@docker01-rubeus:~$
```

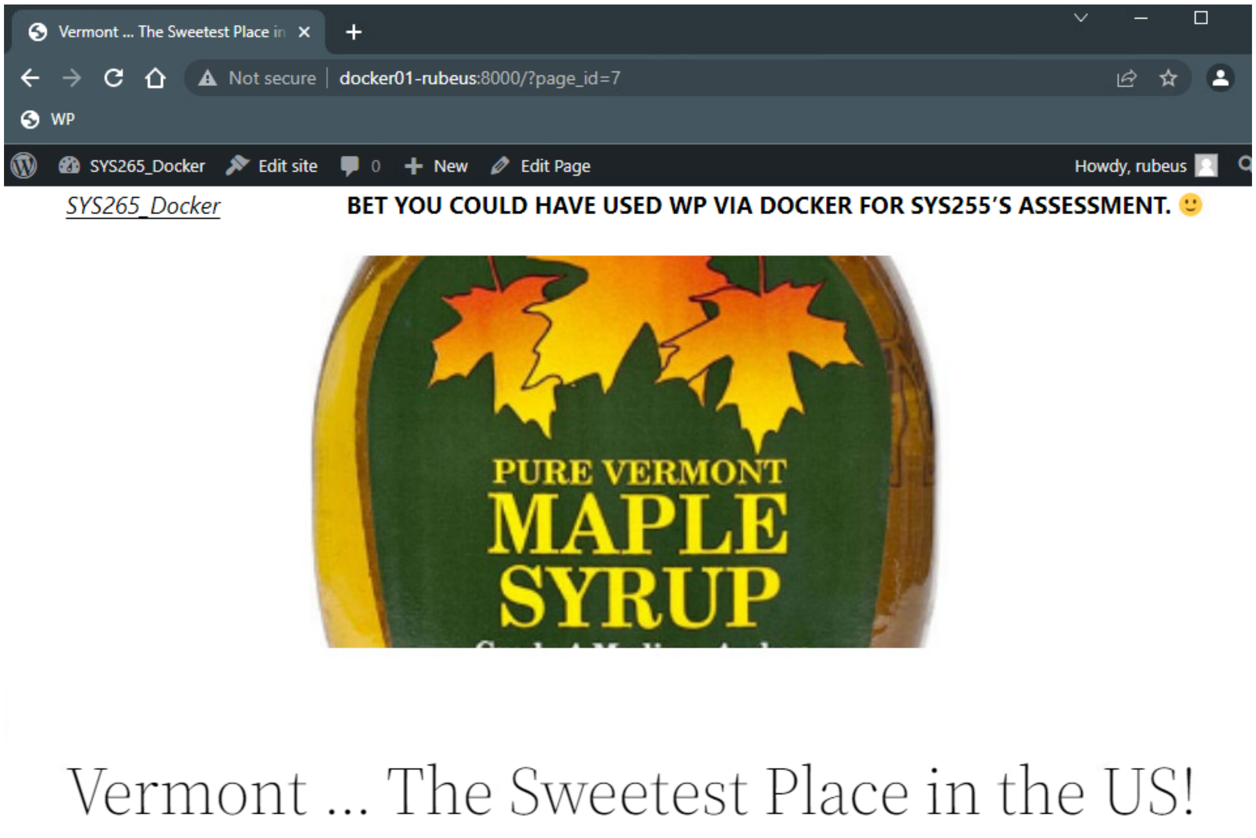
Dockerized Wordpress

In this example, we will use a docker compose file (docker-compose.yml) to identify the attributes of a wordpress installation to include the operating system, software and database dependencies. We will use docker-compose (as opposed to docker run) to bring up the container.

Parse instructions on Quickstart: Compose and WordPress to create and configure a new wordpress image.Tip: There are plenty of related sites to achieve this.

💣Typing a docker-compose.yml file by hand can be an exercise in frustration if this is the first time you've done it. The yml markup parsing is strictly enforced by docker-compose and is very easy to get wrong. Absolutely never use a tab. Figure out how you can copy paste into a file.

```
Deliverable 10.  Provide a screenshot showing a completed Wordpress
installation that contains reference to the course and your name.
You should be accessing it by hostname and not IP address.
```


Vermont ... The Sweetest Place in

Not secure | docker01-rubeus:8000/?page_id=7

WP

SYS265_Docker Edit site 0 New Edit Page Howdy, rubeus

SYS265_Docker BET YOU COULD HAVE USED WP VIA DOCKER FOR SYS255'S ASSESSMENT. 🙂

Vermont ... The Sweetest Place in the US!

Deliverable 11. Provide a link to your tech journal. In addition to your reflection on this lab, Make sure you spend some time on how to:

- configure networking and netplan on your ubuntu system.
- The differences in adding a sudo user as well as
- some of the frequently used docker commands you have been exposed to.

We are raising the bar on tech journal entries. They should actually be useful, accessible and exceptionally well-formatted.