Exercise 3.1: Deploy a New Application

Overview

In this lab we will deploy a very simple **Python** application, test it using Docker, ingest it into Kubernetes and configure probes to ensure it continues to run. This lab requires the completion of the previous lab, the installation and configuration of a Kubernetes cluster.

Working with Python

1. Install python on your master node. It may already be installed, as is shown in the output below.

```
student@ckad-1:~$ sudo apt-get -y install python
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
python is already the newest version (2.7.12-1~16.04).
python set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
```

2. Locate the python binary on your system.

```
student@ckad-1:~$ which python
```

```
/usr/bin/python
```

3. Create and change into a new directory. The Docker build process pulls everything from the current directory into the image file by default. Make sure the chosen directory is empty.

```
student@ckad-1:~$ mkdir app1
student@ckad-1:~$ cd app1
student@ckad-1:~/app1$ ls -l
```

```
total 0
```

4. Create a simple python script which prints the time and hostname every 5 seconds. There are six commented parts to this script, which should explain what each part is meant to do. The script is included with others in the course tar file, though you are encouraged to create the file by hand if not already familiar with the process. While the command shows **vim** as an example other text editors such as **nano** work just as well.

```
student@ckad-1:~/app1$ vim simple.py
```

simple.py

```python
#!/usr/bin/python
## Import the necessary modules
import time
import socket

## Use an ongoing while loop to generate output
```

```python
while True :

## Set the hostname and the current date
  host = socket.gethostname()
  date = time.strftime("%Y-%m-%d %H:%M:%S")

## Convert the date output to a string
  now = str(date)

## Open the file named date in append mode
## Append the output of hostname and time
  f = open("date.out", "a" )
  f.write(now + "\n")
  f.write(host + "\n")
  f.close()

## Sleep for five seconds then continue the loop
  time.sleep(5)
```

5. Make the file executable and test that it works. Use `Ctrl-C` to interrupt the `while` loop after 20 or 30 seconds. The output will be sent to a newly created file in your current directory called `date.out`.

```
student@ckad-1:~/app1$ chmod +x simple.py
student@ckad-1:~/app1$ ./simple.py
```

```
^CTraceback (most recent call last):
  File "./simple.py", line 42, in <module>
    time.sleep(5)
KeyboardInterrupt
```

6. and timedate stamps.

```
student@ckad-1:~/app1$ cat date.out
```

```
2018-03-22 15:51:38
ckad-1
2018-03-22 15:51:43
ckad-1
2018-03-22 15:51:48
ckad-1
<output_omitted>
```

7. Create a text file named `Dockerfile`.

Very Important

The name is important: it cannot have a suffix.

We will use three statements, `FROM` to declare which version of Python to use, `ADD` to include our script and `CMD` to indicate the action of the container. Should you be including more complex tasks you may need to install extra libraries, shown commented out as `RUN pip install` in the following example.

```
student@ckad-1:~/app1$ vim Dockerfile
```

Dockerfile

```
FROM python:2
ADD simple.py /
## RUN pip install pystrich
CMD [ "python", "./simple.py" ]
```

8. Build the container. The output below shows mid-build as necessary software is downloaded. You will need to use **sudo** in order to run this command. After the three step process completes the last line of output should indicate success. Note the dot (.) at the end of the command indicates the current directory.

```
student@ckad-1:~/app1$ sudo docker build -t simpleapp .
```

```
Sending build context to Docker daemon 3.072 kB
Step 1/3 : FROM python:2
2: Pulling from library/python
4176fe04cefe: Pull complete
851356ecf618: Pull complete
6115379c7b49: Pull complete
aaf7d781d601: Extracting [================>         ] 54.03 MB/135 MB
40cf661a3cc4: Download complete
c582f0b73e63: Download complete
6c1ea8f72a0d: Download complete
7051a41ae6b7: Download complete
<output_omitted>
Successfully built c4e0679b9c36
```

9. Verify you can see the new image among others downloaded during the build process, installed to support the cluster, or you may have already worked with. The newly created `simpleapp` image should be listed first.

```
student@ckad-1:~/app1$ sudo docker images
```

```
REPOSITORY                 TAG      IMAGE ID       CREATED         SIZE
simpleapp                  latest   ba54e4910397   6 seconds ago   902MB
k8s.gcr.io/kube-proxy      v1.19.0  4e68534e24f6   7 days ago      117MB
k8s.gcr.io/kube-apiserver  v1.19.0  a595af0107f9   7 days ago      173MB
<output_omitted>
```

10. Use **sudo docker** to run a container using the new image. While the script is running you won't see any output and the shell will be occupied running the image in the background. After 30 seconds use **ctrl-c** to interrupt. The local `date.out` file will not be updated with new times, instead that output will be a file of the container image.

```
student@ckad-1:~$ sudo docker run simpleapp
```

```
^CTraceback (most recent call last):
  File "./simple.py", line 24, in <module>
    time.sleep(5)
KeyboardInterrupt
```

11. Locate the newly created `date.out` file. The following command should show two files of this name, the one created when we ran simple.py and another under `/var/lib/docker` when run via a Docker container.

```
student@ckad-1:~/app1$ sudo find / -name date.out
```

```
/home/student/app1/date.out
/var/lib/docker/overlay2/ee814320c900bd24fad0c5db4a258d3c2b78a19cde
629d7de7d27270d6a0c1f5/diff/date.out
```

12. View the contents of the `date.out` file created via Docker. Note the need for **sudo** as Docker created the file this time, and the owner is `root`. The long name is shown on several lines in the example, but would be a single line when typed or copied.

```
student@ckad-1:~/app1$ sudo tail \
  /var/lib/docker/overlay2/ee814320c900bd24fad0c5db4a258d3c2b78a19cde
629d7de7d27270d6a0c1f5/diff/date.out
```

```
2018-03-22 16:13:46
53e1093e5d39
2018-03-22 16:13:51
53e1093e5d39
2018-03-22 16:13:56
53e1093e5d39
```