Docker Case Study

Problem : Automate Infra allocation for L&D

Requirements:

Requirements:-

1. Dynamic Allocation of Linux systems for users
2. Each user should have independent Linux System
3. Specific training environment should be created in Container
4. User should not allow to access other containers/images
5. User should not allow to access docker command
6. Monitor participants containers .
7. Debug/live demo for the participants if they have any doubts/bug in running applications.
8. Automate container creation and deletion.

So, let us first create container image for the training purpose

Creating the container image:-

1. Create a new container from a base image

```
sudo docker create -it --name docker_test_container ubuntu /bin/bash
```

2. Start the container

```
sudo docker start docker_contain
```

3. Attach to the container

```
sudo docker attach docker_contain
```

4. Install packages required (no need of “sudo”)

```
apt update
apt install vim
apt install gcc
```

5. Create `questions.txt`, `instructions.txt` and save them.

```
touch questions.txt
touch instructions.txt
```

6. Commit the container

```
docker commit a "Apurva Bhatt" d7ef8ee4870f docker_image
```

Now our training container image is ready. Lets allocate it to the users.

Allocating Containers To Users:-

1. The shell script `create_Containers.sh` will automatically create a docker container for every user.

   - `users.txt`

     ```
     Apurva Bhatt
     Emma Watson
     Selena Gomez
     ```

   - `create_Containers.sh`

     ```
     echo n "Please enter the file-names and the usernames: "
     read file
     while read user
         do
           docker create it name $user docker_image /bin/bash
         done < $file
     ```

2. Fill the entries in `users.txt` with usernames and run the shell script `containers_create_script.sh`. This creates a docker container corresponding to each username from `users.txt`.
3. The user can then start using the allocated container by running the `containers_use_script.sh` script.

   - `use_Containers.sh`

     ```
     echo n "Plear enter your username: "
     read name
     docker start $name
     docker attach $name
     ```

Basic tips for monitoring the containers

Monitoring The Containers:-

1. To monitor the containers, use the `containers_monitor_script.sh` script.

   - `containers_monitor_script.sh`

     ```
     echo n "Please enter username of container to be
     monitored: "
     read name
     docker logs f $name
     ```

Basic tips for monitoring the containers

Automating deletion of the containers:-

1. Automate the deletion using the `containers_delete_script.sh` script.

- `containers_delete_script.sh`

```
echo n "Are you sure you want to delete containers of all the usernames?

If yes
enter 'Y' , else enter 'N': "
read option
if [ "$option" == "N" ]
then
    echo n "Give the usernames you want to delete and than enter 'exit'
at the end: "
     while read user
          do
               if [ "$user" != "exit" ]
                    then
                         docker rm $user
               else
                    break
               fi
          done
else
    echo n "Enter the name of file containing usernames: "
      read file
      while read user
      do
              docker stop $user
              docker rm $user
      done < $file

fi
```

2. You can either delete all users or user by name using `sh containers_delete_script.sh -x`

By: Apurva Bhatt (IMT2016010)