Docker Case Study:

IMT2016081

Sri Sumanth Y

Automate infra allocation for L&D

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.

Allocate Different Linux systems for different users:

1)To dynamically allocate the linux system to the users, create a shell script Containers.sh to create docker containers for each specified name.

User.txt

```
A
B
C
```

Containers.sh

```
echo -n "Enter the file name "
read file

While read user
   do
      docker create -it --name $user <Image> /bin/bash
  done < $file

-------------------------
```

2)Run the shell script `Containers.sh` and enter the `User.txt`. This creates a docker container corresponding to each username from that file.

3)The user can then use the container allocated using `Container.sh` script.

- Container.sh
-
  ```
  echo -n "Enter your username: "
  read name
  docker start $name
  docker attach $name
  ```

4)This allows user to enter to his/her allocated Linux system and has only access to the bash of that system.

Automating deletion of the containers

- Automate the deletion using the deleteContainers.sh script.
  - deleteContainers.sh
  -
    ```
    echo -n "Enter 'all' to delete all user containers or enter 'user' to delete a specific user container: "
    read typ
    if [ "$typ" == "all" ]
    then
      echo -n "Enter the user list file: "
      read file
      while read user
        do
          docker rm $user
        done < $file
    else
      echo -n "Enter the username: "
      read name
      docker rm $name
    fi
    ```

- This gives two options ie. to either delete all users containers at once or delete a specific user container.

Note: To run any shell script in the terminal use the following command:

```
sh <shell script>
```

or

```
bash <shell script>
```

Monitoring the container

- One can monitor the participants container using the `monitorContainer.sh` script.
    - monitorContainer.sh

    -
      ```
      echo -n "Enter container name to be monitored: "
      read name
      docker logs -f $name
      ```
- This shows the live display of their bash which helps the participants if they have any doubts/bug in running applications.