Docker Case Study

Problem : Automate Infra allocation for L&D

Requirements

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

Setup Linux containers for users

1. Allocation of a Linux based system can be achieved through a shell script `create_user_container.sh`. Such a script would create a docker container for each user based on a docker image.

- `users.txt`

```
user1
user2
user3
```

- `create_user_container.sh`

```
#! /bin/bash
echo -n "Enter the user list file : "
read file

while read user
  do
    docker create -it --name $user <docker-image> /bin/bash
  done < $file
```

The execution of such a shell script would create a docker container corresponding to each user mentioned in the text file.

2. The user can use the allocated container by using the shell script `use_container.sh`

- `use_container.sh`

```
#! /bin/bash
echo -n "Enter your username : "
read name
docker start $name
docker attach $name
```

Through such a shell script the user is allocated a Linux system. Note that the user has access only to that system.

Monitor the containers

To monitor the activities of a particular user use the shell script `monitor_container.sh`

- `monitor_container.sh`

```
#! /bin/bash
echo -n "Enter the container name which has to ne monitored : "
read name
docker logs -f $name
```

Deleting the containers

Automate the task of deletion of containers through the shell script `delete_container.sh`

- `delete_container.sh`

```
echo -n "Enter the user list file : "
read file

while read user
  do
    docker stop $user
    docker rm $user
  done < $file
```

** Note ** : To execute the shell script, use the following command

```
  sh <shell_script>
```