Docker Case Study - Automate Infra allocation

PROBLEM :

- 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.

Creating the container image:-

- A new container must be created from base image.

  -sudo docker create it name docker_list ubuntu /bin/bash
- Start the container

  -sudo docker start my_container
- Attach to the container

  -sudo docker attach my_container
- Install the required applications using the following commands

  -apt update

- apt install vim
- apt install gcc

- Create questions.txt, instructions.txt and save them.
  - touch questions.txt
  -touch instructions.txt
- Commit the container
  - docker commit a "Shashank" 37f609ba3b38 my_container_image
- our container image is ready

Allocation of container for each user

- To allocate resources for different users on the system we create a bash file with name create_containers.sh that creates a docker container for each specified name
- Touch users.txt
- Vim Users .txt
  - alpha
  - beta
  - gamma

-create_containers.sh

-echo -n “Enter users file:”

-read file

-while read user

-do

-docker create -it –name $user <Docker image>/bin/bash

-done <$file

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

• use_Containers.sh

-echo n "Enter your username: "

-read name

-docker start $name

-docker attach $name

Monitoring participants container

- To monitor the user containers create a bash file monitor_containors.sh

• monitor_Containers.sh

-echo n "Enter username of container to be monitored: "

-read name

-docker logs f $name

Automate container deletion

- create a bash file named delete_container.sh

- delete_containers.sh

    -echo -n “Enter the user file: ”

    -read file

    -while read user

    -do

    -docker stop $user

    -docker rm $user

    -done <$file

- You can delete all users using sh delete_Containers.sh x.

IMT2016058
Shashank Somaraju