

Docker Case Study

Automate Infra Allocation for L&D

Requirements:

1. Users should have independent Linux systems.
2. Dynamic Allocation of Linux systems for users.
3. Automate container creation and deletion.
4. Specific training environments in containers.
5. Disallow access to other containers and images.
6. Disallow access to docker commands.
7. Live demo and debug for participants in case of doubts.
8. Monitor participant containers.

Create the Container Image:

1. First, create a new container from a base image.
`sudo docker create -it --name docker_list ubuntu /bin/bash`
2. Then, start the container.
`sudo docker start docker_contain`
3. Attach to the container.
`sudo docker attach docker_contain`
4. Install the required packages.
`apt update`
`apt install vim`
`apt install gcc`
5. Create text files for questions and instructions.
`touch questions.txt`
`touch instructions.txt`
6. Commit the container
`docker commit -a "Nikhil" 34g608ab3b83 docker_contain_image`

Now the training container image is ready.

Allocate Containers To Users:

1. The script create_containers.sh creates a docker container for every user.

users.txt

```
Nikhil  
Nikhil Sairam  
Tushar
```

create_containers.sh

```
echo -n "Enter name of file with usernames:"  
readfile  
while read user  
do  
    docker create -it --name $user docker_contain_image /bin/bash
```

2. Fill the entries in users.txt with usernames and run create_containers.sh. This will create a docker container for each username in the file.
3. The user can then start using the allocated container by running use_containers.sh.

use_containers.sh

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

Monitor The Containers:

Use monitor_containers.sh to monitor the containers.

monitor_containers.sh

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

Automate Deletion of Containers

1. Automate the deletion using delete_containers.sh.

delete_containers.sh

```
echo n "Do you wish to delete containers of all usernames? If yes  
enter 'Y' , else enter 'N': "  
read option  
if [ "$option" == "N" ]  
then  
    echo n "Give the usernames you want to delete and enter 'exit'
```

```

at the end: "
while read user
do
    if [ "$user" != "exit" ]
    then
        docker rm $user
    else
        break
    fi
done

else
    echo n "Enter name of file containing usernames: "
    read file
    while read user
    do
        docker stop $user
        docker rm $user
    done < $file
fi

```

2. sh delete_containers.sh can be used to delete all users or by username.