| GROUP3 MEMBERS MESUT BUNALDI  MUHAMMED KANDUR  HİKMET TÜTÜNCÜ  ONGUN ALP BABA MUHAMMET BURAK ÖZÇELİK  LEMYE CEREN GÜMÜŞ GÖKTUĞ ALİ AKIN  FERDİ SÖNMEZ  OSMAN ÇETİN  EMRE DURAK  MELİKE SERRA KALYON HASNAIN ALI | | DEPLOY3 DECEMBER 2019 | | --- |  REPORT3 Third meeting:  The group member who did not attend the meeting is muhammed KANDUR.  Docker running trials  Docker and mule  store.docker.com == > shows all images  Docker search "image name" == “” shows all images “” including  Docker I cmd+space 🡺 starts  Sudo docker run codingtony/mule  Mule runtime 3.8.1 I indiriyoruz ; https://developer.mulesoft.com/download-mule-esb-runtime  Documentation in the address :  https://www.slideshare.net/RaviSingh441/how-to-deploy-and-run-mule-in-docker  helloworld.xml  <set-payload  value="#[string:Hello World! It is now: #[server.dateTime]]"/>  sudo docker run -d -p 8181:8181 -v ~/Mule-root/apps:/opt/mule/apps -v ~/Mule-root/conf:/opt/mule/conf -v ~/Mule-root/logs:/opt/mule/logs codingtony/mule  Docker run creates a new container and you can start the command with execute  -d (detach mode) provides workin in the background  -e (environment) We do this to determine the flagini environment.  -p The port is specified on the host before the two points, and the port on the docker will specify which port to work with (- p 8080: 80).  ====================================================  Docker container DOWNLOADING  docker pull kalilinux/kali-linux-docker  ====================================================  Docker container WORKING  sudo docker run -ti ubuntu  docker run -ti kalilinux/kali-linux-docker /bin/bash  -t : flag that assigns a terminal inside our running container.  -i : flag that makes the STDIN available for an interactive connection.  ====================================================  ––––––––––––––––––––––––––––––––––––––––––––––––––––  ====================================================  LISTS OF CONTAINER :  docker container ls -a  LIST OF IMAGES :  docker images  ====================================================  ––––––––––––––––––––––––––––––––––––––––––––––––––––  ====================================================  DELETE THE CONTAINER  docker container rm cc3f2ff51cab cd20b396a061  ====================================================  ––––––––––––––––––––––––––––––––––––––––––––––––––––  ====================================================  DELETE THE ALL CONTAINER :  docker container prune  ====================================================  Docker remove images  docker rmi 231d40e811cd\*  \* = imageID  ====================================================  ––––––––––––––––––––––––––––––––––––––––––––––––––––  ====================================================  WORKING DOCKER AND MULE MACHINE  sudo docker run -d -p 8181:8181 -v ~/Mule-root/apps:/opt/mule/apps -v ~/Mule-root/conf:/opt/mule/conf -v ~/Mule-root/logs:/opt/mule/logs codingtony/mule  ====================================================  ––––––––––––––––––––––––––––––––––––––––––––––––––––  ====================================================  If we want to get information about Images  Docker inspect "image\_name"  To send the image to Docker, we can push it after hub repository opens via hub.docker.com  Docker container also works but in the background we mean docker run -i -d --name hello code ubuntu bash  Docker start -ia my\_ubunt Used to log into a paused container  If we want to create working directory, we use -w option.  Yani docker run -i --rm -v /Users/ceren/deneme:/home/test -w /home/test golang go run main.go  Dockerfile = If we want to create an image by ourselves, we need to edit this file.  First of all we write the image from which we will receive  FROM golang  If we write directly golang golang: perceives in the form of lastest, but we specify the version we want.FROM golang:1.5  If we put a file into it later  COPY ./go\_project /home/test |
| --- | --- | --- |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |