**Dockerising an application:**

* Docker installation commands:

1. Set up Docker's apt repository.

# Add Docker's official GPG key:

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

1. To install the latest version, run:

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

1. Verify that the installation is successful by running the hello-world image:

sudo docker run hello-world

4)**Add your user to the group**: sudo usermod -aG docker ubuntu

* clone the git repo  
   cmd: git clone [https://ghp\_e9G2rvmyVpKyUasJo2VJhy7nXgI0Q32biUKS@github.com/UMENIT-SOLUTIONS-LLP/chatbotrag\_datascience\_python.git](https://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@github.com/UMENIT-SOLUTIONS-LLP/chatbotrag_datascience_python.git" \o "https://ghp_e9g2rvmyvpkyuasjo2vjhy7nxgi0q32biuks@github.com/umenit-solutions-llp/chatbotrag_datascience_python.git" \t "_blank)
* enter into the repo
* Cmd: cd <reponame>
* change to the branch which you want to dockerize
* Cmd: git checkout <branch\_name>
* create a docker file in that branch which you want to dockerize
* Cmd: nano dockerfile (to create and open the dockerfile)
* build the docker image for the above docker file
* Cmd: docker build -t <image name> .
* run the docker container
* Cmd: docker run --name <give the container name> -p 8080:8080 -it <give your image name>
* To access the application
* Edit inbound rules in the ec2 security groups before accessing the application,

After that in open a new tab :http://<public ip of instance>:<potr number>

* To check images cmd: docker images
* To check running containers cmd: docker ps
* To check all containers cmd : docker ps -a
* To start container cmd : docker start<container ID/name>

**Pushing the docker image from the server to the docker hub:**

* Step 1: Log in to Docker Hub

Cmd : Docker login

Enter the url in browser and and paste the code you got in server, login with docker hub credentials

* Step 2: Tag the Docker Image

Cmd:docker tag <local-image-name> <docker-hub-username>/<repository-name>:<tag>

* Step 3: Push the Docker Image

Cmd:docker push <docker-hub-username>/<repository-name>:<tag>

* Step 4: Verify the Image on Docker Hub  
   cmd:check in the docker hub in the repository

**Pulling Image from docker hub to azure ACR:**

Create a resourse group

Create a repository in container registry  
 cmd:az acr create --resource-group <resource-group-name> --name <acr-name> --sku Basic

cmd:Docker login

* Step 1:Pull the Docker Image from Docker Hub  
  cmd : docker pull <DOCKER\_HUB\_IMAGE>  
  ex: docker pull umenit/umenbot-private:umenbot\_python
* Step 2:Tag the Docker Image for ACR  
  cmd:docker tag <DOCKER\_HUB\_IMAGE> <ACR\_LOGIN\_SERVER>/<REPOSITORY\_NAME>:<TAG>  
   Ex: docker tag umenit/umenbot-private:umenbot\_python umenbotpythonds.azurecr.io/umenbotpythonds:umenbot\_python
* Step 3 :Log in to Azure Container Registry  
  cmd: az acr login --name <REPOSITORY\_NAME>

Ex: az acr login --name umenbotpythonds

* Step 4: Push the Docker Image to ACR  
  cmd: docker push <ACR\_LOGIN\_SERVER>/<REPOSITORY\_NAME>:<TAG>  
  ex: docker push umenbotpythonds.azurecr.io/umenbotpythonds:umenbot\_python
* Step 5: Verify the Image in ACR  
  cmd: az acr repository list --name <ACR\_NAME> --output table  
  ex: az acr repository list --name umenbotpythonds --output table

**Push Image from a Server to ACR:**

-