**Docker Container For IPSP**

1. **Install Docker on Linux VM**

sudo apt-get update

sudo apt-get install \

ca-certificates \

curl \

gnupg \

lsb-release

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive- keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io

sudo systemctl status docker

docker ps

1. **Create a folder for Creating docker in directory**

mkdir ~/my-dotnet-app

cd ~/my-dotnet-app

nano Dockerfile

[

FROM mcr.microsoft.com/dotnet/aspnet:6.0

WORKDIR /app

COPY . .

ENTRYPOINT ["dotnet", "IPSP.Server.dll"]

]

Save and Exit the Dockerfile

1. **Copy the update main server file and paste it in ipsp server directory**

After writing the Dockerfile, press CTRL + O to save the file, and CTRL + X to exit.

Add Your Application Files

[Now, copy your .NET application files to the /app directory on your Ubuntu VM. If your files are in a different directory, you can use the scp command to transfer them from your local machine or another server]

For example, if your app file is located at /home/gama/ipsp/\* use:

cp -r /home/gama/ipsp/\* ~/ my-dotnet-app /

1. **Steps to Create a Docker Network(Optional)**

docker network ls

docker network create --subnet=192.168.200.0/24 docker\_network

docker network inspect [network\_name/ID]

1. **Build the Docker Image**

Once you’ve created the Dockerfile and added your application files, build the Docker image.

1. Navigate to the directory where the Dockerfile is located:

cd ~/my-dotnet-app

1. Build the Docker image:
2. docker build -t my-dotnet-app .
3. **Run the Docker Container**

docker run -d --net docker\_network --ip 192.168.200.51 -p 9095:9095 -e IPSP\_Encryption\_Key="GamaEvtet\_2021\_IPSP@PasswordEncryption" -e IPSP\_SSL\_Key="biju@1234" -e POSTGRES\_HOST="192.168.100.88" -e POSTGRES\_USER="ch-db" -e POSTGRES\_PASSWORD="admin@1234" -e POSTGRES\_DB="IPSPClient" my-dotnet-app

docker ps [Check if the container is running]

docker logs [docker ID]

1. **Test Your Application**

<https://192.168.100.51:9095/swagger/index.html> [ 192.168.200.51 IP address for within docker communication and exposing ip address always be Vm machine Ip address]

**Docker Containers push to the DockerHub/GitHub**

**1)Push the images to a container registry:** If you have access to a Docker registry like Docker Hub, Google Container Registry (GCR), or a private registry, push your images there. Kubernetes can pull them from the registry when deploying.

Tag your Docker image to your repository:

docker tag <your\_image> <your\_registry>/your\_image:tag

docker push <your\_registry>/your\_image:tag

If you're using Docker Hub

docker tag <your\_image> your\_dockerhub\_username/your\_image:tag

docker push your\_dockerhub\_username/your\_image:tag

**2) Steps to Push Docker Image to Docker Hub:**

docker tag my-dotnet-app saif24k6/ipsp:latest

{docker tag <your\_image> your\_dockerhub\_username/your\_image:tag}

**3) Login to Docker Hub:**

docker login

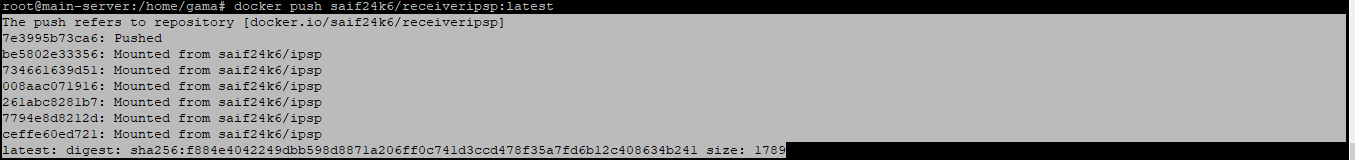
**4)Download DockerHub in your laptop:**

Install and login to your account and when I pushed docker login so it requires authentication code which you get from <https://docker.com/activate> and put the code which Linux machine gives you after pushing docker login command and you can see successfully login from your vm machine to push the tag image

**5)** **Push the Image to Docker Hub**

docker push saif24k6/ipsp:latest

{docker push your\_dockerhub\_username/your\_image:tag}



**Alternate:**

**6)** **Copy the Docker images to the worker nodes:** If you don't want to push the images to a registry, you can manually copy the Docker images to your Kubernetes worker nodes.

docker save -o my\_image.tar <your\_image>

[example : docker save -o my\_receiverimage.tar receiveripsp]

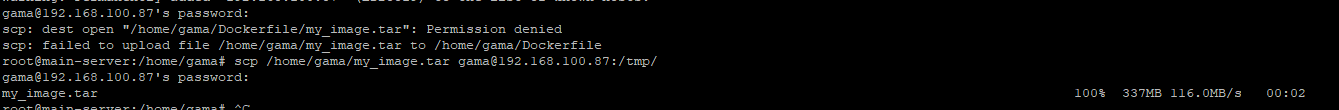
sudo chmod 777 /source/path – given permission to transfer file

scp /source/path/\*\*\*.tar dest machine username@ip addr:/dest/path/

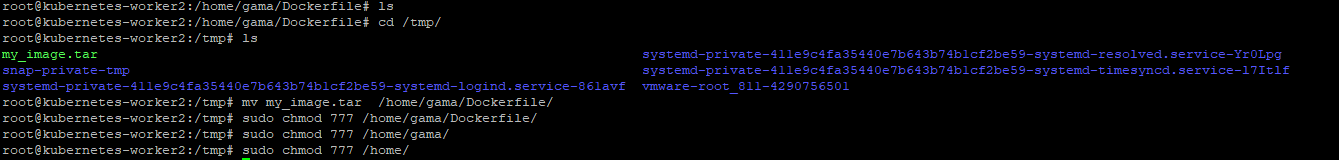
[example : scp /home/gama/my\_receiverimage.tar [gama@192.168.100.86:/home/gama/Dockerimage](mailto:gama@192.168.100.86:/home/gama/Dockerimage)]

[if there is error of permission denied so transfer file to tmp folder and go to the worker node move file from temp to your folder and give permission to read-write so that nect time directly send to your directory]

**Source VM:**



**Destination VM:**



**Load the image on each worker node:**



[if docker is not running because of some reason restart with docker container ID]