Docker Basic Commands :

Docker Registry :

A Docker registry is a central repository for storing and distributing Docker images. It serves as a place where Docker images can be pushed, pulled, and managed by users and automated systems. Docker registries play a crucial role in the Docker ecosystem by facilitating the sharing and deployment of containerized applications. Here are some key aspects of Docker registries:

Key Features of Docker Registries

1. **Image Storage**: Docker registries store Docker images, which are the packaged applications and their dependencies.
2. **Distribution**: Docker registries provide a means for distributing images to multiple servers or environments.
3. **Versioning**: Docker registries support versioning of images, allowing users to track and manage different versions of their applications.
4. **Access Control**: Docker registries can enforce access control policies, ensuring that only authorized users have permissions to push, pull, or manage images.
5. **Replication**: Docker registries can replicate images across multiple nodes or data centers, improving availability and reliability.
6. **Integration**: Docker registries integrate with various CI/CD pipelines, build systems, and deployment tools, enabling automated workflows for building, testing, and deploying containerized applications.

Examples of Docker registry : Docker Hub, Amazon Elastic Container Registry (ECR), Google Container Registry (GCR) , Azure Container Registry (ACR).

Docker Basic Commands :

`[ImageName]` : Name of the image as you find it in the registry.

`[containerName]` : It is the name of the running/executing image.

- You start the execution of the container with the name of the Image and you interact with the running/executing image with the container Name.

```
# To view all the downloaded Images
docker images

#Get image info (Debugging)
docker image inspect [ImageName]

#Pull / Download an image from a registry
docker pull [ImageName]

#Run Containers/Execute the image in the memory as a containe
r
#If the image is not present inside the local cache, then it
#will be download it automatically
docker run [ImageName]

#Run the container in background(Get back the terminal acces
s)
docker run -d [ImageName]

#To map the port of Host machine with the container port
docker run --publish [HostPort]:[containerPort] --name webser
ver nginx

                      OR
docker run -p[HostPort]:[containerPort] [imageName]

#To map the port of Host machine with the container port and
run it in the detach mode
```

```
docker run -p[HostPort]:[containerPort] -d [imageName]

#Runs the docker which are the stopped containers
docker start [containerName]

#List running containers
docker ps

#List running and stopped containers
docker ps -a

#Stops a container but it is still in the memory
docker stop [containerName]
        OR
docker stop [containerID]

#Kills containers
docker kill

#To restart a running or stopped container
docker restart [containerName]
```

Command To Give A Custom Name To The Container:

```
docker run --name [containerName] [ImageName]

#To map the port of Host machine with the container port and
run it with a custom name
docker run -d -p[HostPort]:[ContainerPort] --name [CustomName
OfContainer] [ImageName]

                                    OR
```

```
docker run -p[HostPort]:[ContainerPort] --name [CustomNameOfC
ontainer] -d [ImageName]
```

Limiting The CPU And Memory Resources :

```
#Max Memory
docker run --memory="256m" nginx

#Max CPU
docker run --cpus=".5" nginx
```

Attach File/Bash To The Containers :

```
#Attach shell
docker run -it nginx -- /bin/bash

#Attach Powershell
docker run -it -- microsoft/powershell:nanoserver pwsh.exe

#Attach to a running container
#Open a bash shell in the container
docker container exec -it [containerName] -- bash

docker exec -it <container-id> /bin/sh
                 OR
docker exec -it <container-id> /bin/bash
```

Cleaning The Containers :

```
#Removes Stopped containers (The containers must be in stoppe
d state) .
#This command is also used to remove the stopped containers f
rom the memory
docker rm [containerName]

#Removes all the stopped containers
docker rm $(docker ps -a -q)

#Lists Images
docker images

#To delete the images
docker rmi [ImageName]

#Remove all images not in use by any containers
docker system prune -a
```

Docker Network :

```
#To create a docker network
docker network create <network-name>

#To view all the docker network
docker network ls

#Provides detailed information about the specified network
#including its configuration and the containers connected to it
docker network inspect <network-name>

#Connects an existing container to the specified network.
docker network connect <network-name> <container-name>
```

```
#Disconnects a container from the specified network.
docker network disconnect <network-name> <container-name>

#Removes the specified network.
#Note that the network must not have any active containers conne
docker network rm <network-name>

#Removes all unused networks, freeing up resources.
docker network prune
```