Docker Commands Notes

Commands for Images:

- docker pull image_name - Pull an image (e.g., docker pull ubuntu)
- docker images or docker image ls - List all docker images
- docker rmi image_name/image_id - Delete an image from docker host
- docker push image_name - Upload a docker image to Docker Hub
- docker tag image_name - Rename an image
- docker commit container_name/container_id new_image_name - Build an image from a customized container
- docker build -t new_image_name . - Create an image from a Dockerfile
- docker search image_name - Search for a Docker image
- docker system prune -a - Delete all unused images

Commands for Containers:

- docker run image-name - Run a container
- docker container ls or docker ps - List all running containers
- docker ps -a - List running and stopped containers
- docker start container_name/container_id - Start a container
- docker stop container_name/container_id - Stop a container
- docker restart container_name/container_id - Restart a container
- docker restart -t 10 container_name/container_id - Restart a container after 10 minutes
- docker rm container_name/container_id - Delete a stopped container
- docker rm -f container_name/container_id - Force delete a running container
- docker stop $(docker ps -aq) - Stop all running containers
- docker restart $(docker ps -aq) - Restart all running containers
- docker rm -f $(docker ps -aq) - Remove all running and stopped containers
- docker logs container_name/container_id - View container logs
- docker port container_name/container_id - View container ports
- docker inspect container_name/container_id - View detailed container information
- docker attach container_name/container_id - Attach to a running container shell
- docker exec -it container_name/container_id command - Execute a command inside a container
- docker exec -it container_name/container_id bash - Launch a bash shell inside a container

Flags:

- --name - Assign a name to a container
- -it - Open an interactive terminal in a container
- -d - Run a container in detached mode
- -e or --env - Pass environment variables to a container
- -p - Port mapping
- -P - Automatic port mapping
- -v - Attach a volume to a container

Dockerfile Keywords:

- FROM - Specify base image
- MAINTAINER - Define the author or organization
- CMD - Default command when the container starts
- ENTRYPOINT - Default process to execute when the container starts
- RUN - Execute Linux commands (used for installing software)
- USER - Define default user
- WORKDIR - Define default working directory
- COPY - Copy files from the host to the container
- ADD - Copy files (also supports remote URLs and ZIP extraction)
- ENV - Define environment variables
- EXPOSE - Define internal container ports
- VOLUME - Define default volumes
- LABEL - Assign metadata labels to a container

Pushing to Docker Hub:

1. Sign in to Docker Hub and create a repository.
2. Log in to Docker from the terminal: docker login
3. Enter Docker Hub credentials.
4. Tag the image: docker tag imagename username/repositoryname:tag
5. List images: docker images
6. Push the image: docker push username/repositoryname:tag

Volumes:

- docker volume create vol-name - Manually create a volume

- docker run -it --name demo -v vol-name:/mylogs img-name - Create and attach a volume

Docker Compose:

- apt update && apt install docker-compose -y - Install Docker Compose
- Create a YAML file for container configuration
- docker-compose -f compose.yml config - Validate YAML syntax
- docker-compose up -d - Start containers in detached mode
- docker-compose down - Stop and remove containers

Docker Swarm:

- sudo hostname new-name - Change the hostname
- docker swarm init - Initialize Docker Swarm
- docker swarm join-token manager - Generate a token for manager nodes
- docker swarm join-token worker - Generate a token for worker nodes
- docker info | grep -i swarm - Check swarm status
- docker node ls - List swarm nodes
- docker network ls - List networks

High Availability & Fault Tolerance:

- docker service create --name <service_name> <image_name> - Create a service
- docker service create --name first --replicas=3 -p 31000:80 nginx - Deploy a service with 3 replicas
- docker service ls - List services
- docker service ps first - Check service tasks
- docker ps -a - List all containers
- docker node ls - List swarm nodes
- watch docker node ls - Monitor swarm nodes

Scaling Services:

- docker service scale first=5 - Scale up to 5 replicas
- docker service scale first=2 - Scale down to 2 replicas

Rolling Updates & Rollbacks:

- docker service create --name myweb -p 32000:80 nginx:1.17 - Deploy a service
- docker service update myweb --image nginx:1.18 - Perform rolling update
- docker service ps myweb - Check service status

- docker service rollback myweb - Roll back to the previous version

Managing Services & Nodes:

- docker service rm myweb - Remove a service
- docker service ls - List services
- docker node update --availability=Drain docker-manager-1 - Drain a manager node
- docker node rm hostname/id - Remove a node from the cluster
- docker swarm leave - Leave the swarm (execute on the node you want to remove)
- docker node promote worker1 - Promote a worker to manager
- docker node demote manager2 - Demote a manager to worker