Docker cheatsheet

SYED DADAPEER

WHAT IS DOCKER?

Docker allows you to package and run an application in a container, which is a loosely isolated environment. Because of the isolation and security, you can run multiple containerson a single host at the same time. Containers are lightweight and contain everything required to run the application, eliminating the need to rely on what is already installed on the host. You can easily share containerswhile working, and you can ensure that everyone with whom you share gets thesame container that works in the same way.

DOCKER INSTALLATION

Docker Desktop is available for Mac, Linux and Windows
https://docs.docker.com/desktop

View example projects that use Docker
https://github.com/docker/awesome-compose

Check out our docs for information on using Docker
https://docs.docker.com

DOCKER IMAGES

Docker images are a lightweight, standalone, executable package of sofware that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Build an Image from a Dockerfle
$ docker build -t <image_name>

Build an Image from a Dockerfle without the cache
$ docker build -t <image_name> . –no-cache

List local images
$ docker images

Delete an Image
$ docker rmi <image_name>

Remove all unused images
$ docker image prune

DOCKER HUB

Docker Hub is a service provided by Docker for fnding and sharing container images with your team. Learn more and fnd images at: https://hub.docker.com.

Login into Docker
$ docker login -u <username>

Publish an image to Docker Hub
$ docker push <username>/<image_name>

Search Hub for an image
$ docker search <image_name>

Pull an image from a Docker Hub
$ docker pull <image_name>

Pulsh an image to Docker Hub
$ docker push <image_name>

Logout from Docker Hub
$ docker logout

DOCKER GENERAL COMMANDS

Docker Hub is a service provided by Docker for fnding and sharing container images with your team. Learn more and find images at:
https://hub.docker.com

Start the docker daemon:
$ docker -d

Get help with Docker. Can also use –help on all subcommands:
$ docker --help

Display system-wide information
$ docker info

DOCKER CONTAINERS

A container is a runtime instance of a docker image. A container will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Create and run a container from an image, with a custom name:
$ docker run --name <container_name> <image_name>

Run a container with and publish a container's port(s) to the host.
$ docker run -p <host_port>:<container_port> <image_name>

Run a container in the background
$ docker run -d <image_name>

Start or stop an existing container:
$ docker start|stop <container_name> (or <container-id>)

Remove a stopped container:
$ docker rm <container_name>

Open a shell inside a running container:
$ docker exec -it <container_name> sh

Fetch and follow the logs of a container:
$ docker logs -f <container_name>

To inspect a running container:
$ docker inspect <container_name> (or <container_id>)

To list currently running containers:
$ docker ps

List all docker containers (running and stopped):
$ docker ps --all

View resource usage stats
$ docker container stats

SYED DADAPEER

docker commands essentials

```
## Docker Version and Information
docker --version                          # Show Docker version
docker info                               # Display system-wide information about Docker

## Working with Containers
docker ps                                 # List running containers
docker ps -a                              # List all containers (running and stopped)
docker run <image>                        # Run a container from an image
docker run -d <image>                     # Run a container in detached mode (in the
background)
docker run --name <name> <image>          # Run a container with a custom name
docker stop <container>                   # Stop a running container
docker start <container>                  # Start a stopped container
docker restart <container>                # Restart a container
docker rm <container>                     # Remove a stopped container
docker exec -it <container> /bin/bash     # Access a running container's shell

## Managing Images
docker images                             # List all Docker images
docker pull <image>                       # Pull an image from Docker Hub
docker build -t <name>:<tag> <path>       # Build an image from a Dockerfile
docker rmi <image>                        # Remove a Docker image
docker tag <image> <new_name>:<tag>       # Tag an image with a new name and/or tag
docker push <name>:<tag>                  # Push an image to a Docker registry

## Docker Networks
docker network ls                         # List all Docker networks
docker network create <name>              # Create a new Docker network
docker network inspect <network>          # Display details about a Docker network
docker network connect <network> <container>   # Connect a container to a network
docker network disconnect <network> <container>  # Disconnect a container from a network
docker network rm <network>               # Remove a Docker network
```

SYED DADAPEER

```
## Docker Volumes
docker volume ls                          # List all Docker volumes
docker volume create <name>               # Create a new Docker volume
docker volume inspect <volume>            # Display details about a Docker volume
docker run -v <volume>:/path <image>      # Attach a volume to a container
docker volume rm <volume>                 # Remove a Docker volume

## Viewing Container Logs
docker logs <container>                   # View logs of a container
docker logs -f <container>                # Follow logs of a container (real-time output)

## Inspecting Containers and Images
docker inspect <container/image>          # Display detailed information about a container or
image
docker stats                              # Display resource usage statistics for containers

## Container Export and Import
docker export <container> > <file.tar>    # Export a container's filesystem as a tar archive
docker import <file.tar> <image_name>     # Import a tar archive as a Docker image

## Docker Compose (if installed)
docker-compose up                         # Start all services defined in docker-compose.yml
docker-compose down                       # Stop and remove containers, networks, images, and
volumes
docker-compose logs                       # View logs of all services defined in docker-
compose.yml
docker-compose ps                         # List containers managed by Docker Compose

## Cleaning Up Docker
docker system prune                       # Remove unused data (containers, images, networks,
and volumes)
docker container prune                    # Remove all stopped containers
docker image prune                        # Remove unused Docker images
docker volume prune                       # Remove unused Docker volumes
docker network prune                      # Remove unused Docker networks
```