|  |  |
| --- | --- |
| GENERAL DOCKER CHEAT SHEET (Ref: https://spacelift.io/blog/docker-commands-cheat-sheet) | |
| Docker host IP: host.docker.internal or /sbin/ip route|awk '/default/ { print $3 }' | |
| docker version | Displays detailed info about your Docker CLI and daemon versions. |
| docker system info |  |
| docker help |  |
| docker <command> --help |  |
| Build Images |  |
| docker build . |  |
| docker build -t example:latest | Build the dockerfile and tag it as example:latest |
| docker build -f <docker-file> |  |
| docker -t ex:tag -f <docker-file>  --build-arg foo=bar | Build docker file, pass-in build arguments and docker definition file |
| Docker build –pull . | Instructs Docker to pull updated versions of the images referenced in FROM instructions in your Dockerfile, before building your new image. |
| docker build --quiet . | Quietly build docker |
| RUNNING CONTAINER |  |
| docker run <tag> --rm [-it][-d]  <arguments> --name <name> --env foo=1  -p 8080:80  -v <host-dir>:<container-dir>  --network my-net  --restart unless-stopped  --privileged | --rm – remove container when it existed  -d detach your terminal after  -it attaches to terminal  --name gives it a name  --env creates a environment  -p host-port:container-port |
| MANAGE CONTAINER |  |
| Docker ps -a | List all containers |
| Docker attach <container> | Attach your terminal to the foreground process of the container with the ID or name <container>  **TO DETACH PRESS CTRL+P and CTRL+Q** |
| Docker commit <container> newtag:latest | Commit the running image to new tag |
| Docker inspect <container> | Obtain all the information about a container, in JSON format. |
| Docker kill <container> | Send a SIGKILL signal to the foreground process running in a container, to force it to stop |
| Docker rename container <new-name> |  |
| Docker pause <container> | docker unpause <container> works to unpause |
| Docker stop <container> |  |
| Docker rm [-f] container | Delete container by its ID -f force |
| COPY to and FROM CONTAINER |  |
| Docker cp ex.txt container:/data | Copy ex.txt from host to the container |
| Docker co container:/data/ex.txt /tmp | Copy file from running container to host |
| EXECUTE COMMANDS in CONTAINER |  |
| docker exec [-it] <container> <cmd>  Ex: docker exec -it note1 bash | Execute a command in container [-it] will provide a interactive shell |
| ACCESS CONTAINER LOGS |  |
| Docker logs <container> [--follow] [-n 10] | This command streams the existing log output from a container into your terminal window. [--follow] flag will continue to log  [-n 10] get last 10 lines |
| Docker stats <container> | Shows CPU memory usage etc |
| IMAGE management |  |
| Docker images -ls |  |
| Docker rmi <image> | Delete the image |
| Docker tag <image> ex-tag:latest | Add a tag ex-tag:latest to the image |
| PUSH PULL |  |
| Docker push ex.com/user/image:latest | Push an image from your Docker host to a remote registry. The image is identified by its tag, which must reference the registry. |
| Docker pull ex.com/user/image:latest | Manually pull the image. When the image’s tag omits a registry URL, the Docker Hub registry will be used as the default. |
| MANAGE NETWORK |  |
| docker create network my-network | Creates my-network that can be used to in docker run command – by default it creates a bridge network |
| docker create network my-network -d host | [-d host] flag will create a host network |
| Docker network connect <network>  <container> | Connect container to existing network |
| Docker network disconnect <network>  <container> |  |
| Docker network rm <network> | Removes the network |
| Docker network ls | List all the network |
| MANAGE VOLUME |  |
| Docker volume create my-volume | Creates a new volume called my-volume |
| Docker volume -ls | List volumes present in your host |
| Docker volume rm | Deletes a volume and destroys all data in it – Volume must not be in use by any other container |
| CONFIGURATION CONTEXT |  |
| docker context create my-context --host=tcp://host:2376,ca=~/ca-file,cert=~/cert-file,key=~/key-file | Create a new context called my-context to connect to a specified Docker host |
| docker context update <context> | Modify the configuration of a named context; the command accepts the same arguments as docker context create |
| docker context ls | List the contexts available in your Docker config file |
| docker context use <context> | Switch to a named context. Subsequent docker commands will be executed against the Docker host configured in the newly selected context. |
| docker context rm <context> | Deletes the context |
| CREATE SBOMS (https://docs.docker.com/engine/sbom/ | |
| docker sbom ex-image:latest | Produce an SBOM for the image tagged example-image:latest. The SBOM will be shown in your terminal. |
| docker sbom ex-image:latest --output m.txt | Same as aboive – but output us saved in m.txt |
| docker sbom ex-image:latest --format spdx-json | Produce an SBOM in a standard machine-parseable format, such as SPDX (spdx-json), CycloneDX (cyclonedx-json), or Syft JSON (syft-json). |
| SCAN For Vulnerabilities |  |
| docker scan example-image:latest | – Scan for vulnerabilities in the image tagged example-image:latest. The results will be shown in your terminal. |
| docker scan ex-image:latest --file Dockerfile | – The --file argument supplies the path to the Dockerfile that was used to build the image. When the Dockerfile is available, more detailed vulnerability information is produced. |
| docker scan example-image:latest --severity high | Only report vulnerabilities that are high severity or higher. The --severity flag also supports low and medium values |
| Docker HUB Account |  |
| Docker login |  |
| Docker logout |  |
| Docker search nginx | Searches Docker Hub for images matching the supplied search term (nginx, in this example). |
| CLEANING UP RESOURCES |  |
| docker system prune [-a] [--volumes] | Removes unused data, including dangling image layers (images with no tags).  [-a] – option Extends the prune process by deleting all unused images, instead of only dangling ones.  [--volumes] prune volume and will delete any volumes that aren’t used by a container. |
| Docker image prune [-a] |  |
| Docker network prune |  |
| Docker volume prune |  |
| Docker system df | Reports your Docker installation’s total disk usage. |