Docker Cheatsheet

Tutorial Series

Get started with Docker: https://docs.docker.com/engine/getstarted/

Installation

Linux

Install script provided by Docker:

```
curl -sSL https://get.docker.com/ | sh
```

Or see Installation instructions for your Linux distribution.

Mac OS X

Download and install Docker For Mac

Create Docker VM with Docker Machine

You can use Docker Machine to

- Install and run Docker on Mac or Windows
- Provision and manage multiple remote Docker hosts
- Provision Swarm clusters

A simple example to create a local Docker VM with VirtualBox:

```
docker-machine create --driver=virtualbox default
docker-machine ls
eval "$(docker-machine env default)"
```

Then start up a container:

```
docker run alpine echo "hello-world"
```

That's it, you have a running Docker container.

METRICS, EVENTS and LOGS

START YOUR FREE TRIAL

Sematext is Docker monitoring and logging certified partner

Container lifecycle

- Create a container: `docker create imageName`.
- Create and start a container in one operation: `docker run imageName`
  - Remove the container after it stops `--rm`: `docker run --rm alpine ls /usr/lib`
  - Attach the container stdin/stdout to the current terminal use `-it`: `docker run -it ubuntu bash`
  - To mount a directory on the host to a container add `-v` option, eg `docker run -v $HOSTDIR:$DOCKERDIR imageName`
  - To map a container port use `-p $HOSTPORT:$CONTAINERPORT`: `docker run -p 8080:80 nginx`
  - To run the container in background use `-d` switch: `docker run -d -p 8080:80 nginx`
  - Set container name: `docker run --name myContainerName imageName`
  - Example to run nginx web server to serve files from html directory on port 8080:

```
docker run -d -v $(pwd)/html:/usr/share/nginx/html -p 8080:80 --name myNginx
nginx
# access the webserver
curl http://localhost:8080
```

- In some cases containers need extended privileges. Add privileges with the `--cap_add` switch, eg `docker run --cap-add SYS_ADMIN imageName`. The flag `--cap_drop` is used to remove priviledges
- Rename a container: `docker rename name newName`
- Delete a container: `docker rm containerID`
  - Delete all unused containers: `docker ps -q -a | xargs docker rm`
  - Remove the volumes associated with a container: `docker rm -v containerName`
- Update container resource limits: `docker update --cpu-shares 512 -m 300M`
- List running containers: `docker ps`
- List all containers: `docker ps -a`
- List all container IDs: `docker ps -q -a`

Starting and stopping containers

- Start a container: `docker start containerName`
- Start a container: `docker stop containerName`
- Restart a container: `docker restart containerName`
- Pause a container ("freeze"): `docker pause containerName`
- Unpause a container: `docker unpause containerName`
- Stop and wait for termination: `docker wait containerName`
- Kill a container (sends SIGKILL): `docker kill containerName`

Executing commands in containers and apply changes

- Execute a command in a container: `docker exec -it containerName command`
- Copy files or folders between a container and the local filesystem: `docker cp containerName:path localFile` or `docker cp localPath containerName:path`
- Apply changes in container file systems: `docker commit containerName`

Docker networks

- List existing networks: `docker network ls`
- Create a network: `docker network create netName`
- Remove network: `docker network rm netName`
- Show network details: `docker network inspect`
- Connect container to a network: `docker network connect networkName containerName`
- Disconnect network from container: `docker network disconnect networkName containerName`

Logging and monitoring

Logging on Docker could be challenging - check Top 10 Docker Logging Gotchas.

- Show container logs: `docker logs containerName`
- Show only new logs: `docker logs -f containerName`
- Show CPU and memory usage: `docker stats`
- Show CPU and memory usage for specific containers: `docker stats containerName1 containerName2`
- Show running processes in a container: `docker top containerName`
- Show Docker events: `docker events`
- Show storage usage: `docker system df`
- To run a container with a custom logging driver i.e., to syslog use `docker run \ --log-driver syslog --log-opt syslog-address=udp://syslog-server:514 \ alpine echo hello world`.
- Use Sematext Docker Agent for log collection. Create Docker Monitoring App & Logs App in Sematext Cloud UI to get required tokens. Start collecting all container metrics, host metrics, container logs and Docker events:

```
docker run -d --name sematext-agent-docker \
-e SPM_TOKEN=YOUR_SPM_TOKEN
-e LOGSENE_TOKEN=YOUR_LOGSENE_TOKEN \
-v /var/run/docker.sock:/var/run/docker.sock \
sematext/sematext-agent-docker
```

The command above will collect all container metrics, host metrics, container logs and Docker events.

Exploring Docker information

- Show Docker info: `docker info`
- List all containers: `docker ps -a`
- List all images: `docker image ls`
- Show all container details: `docker inspect containerName`
- Show changes in the container's files: `docker diff containerName`

Manage Docker images

- List all images: `docker images`
- Search in registry for an image: `docker search searchTerm` pulls an image from registry to local machine
- Pull image from a registry: `docker pull imageName`
- Create image from Dockerfile: `docker build`
- Remove image: `docker rmi imageName`
- Export container into tgz file: `docker export myContainerName -o myContainerName`
- Create an image from a tgz file: `docker import file`

Data cleanup

Data Management Commands:

- Remove unused resources: `docker system prune`
- Remove unused volumes: `docker volume prune`
- Remove unused networks: `docker network prune`
- Remove unused containers: `docker container prune`
- Remove unused images: `docker image prune`

Full-stack visibility. Actionable insights. Single pane of glass.