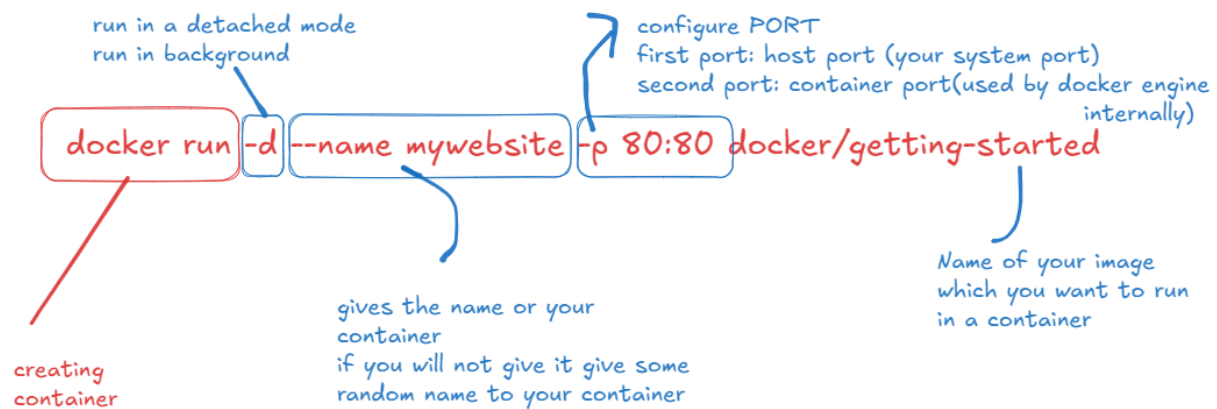
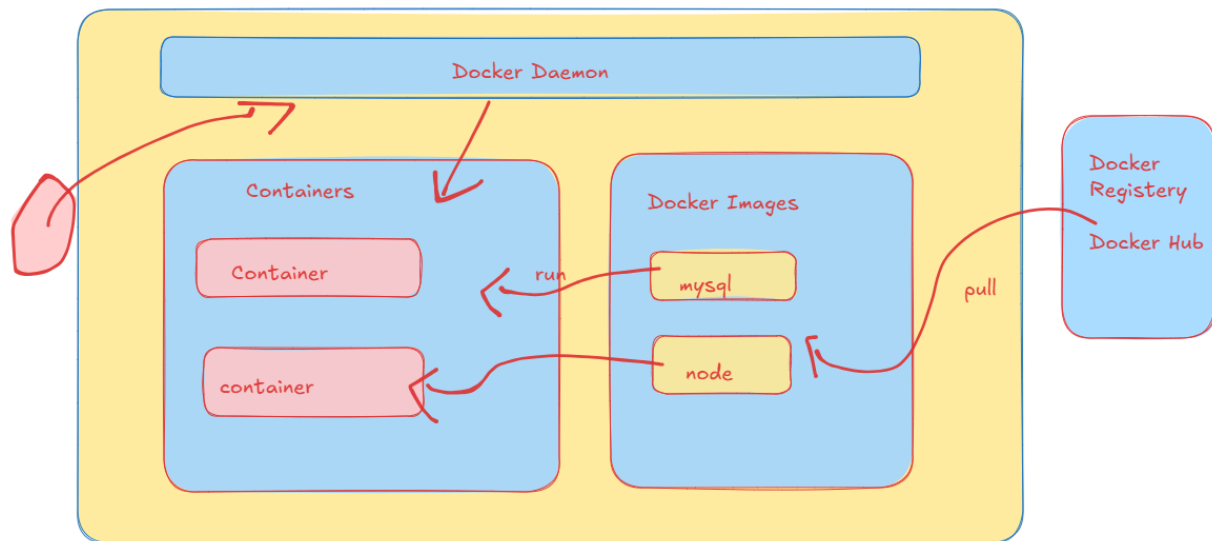


Docker



Docker Version: `docker --version`

Docker version: `docker version` | (if you can see both client + server responses means docker engine is running)

Check Available Images

> `docker images`

Let's Pull One Image named hello-world

>docker pull hello-world

verify the pulled images:

>docker images

let's this image

>docker run hello-world

Let's Check the Containers

>docker ps (it will not show you anything because your image executes shown the output and it exited)

PS is showing up(running) containers but we want to all containers

>docker ps -a (you can see hello-world in exited status)

Remove Exited container

>docker rm name-of-container

Delete the image which is pulled

>docker rmi name-of-image

>docker rmi hello-world

Verify Images: docker images

I want to run one image which using some server to run and we can see the output in browser.

> docker run -d -p 8082:80 --name my-website docker/getting-started

docker run: creates container

-d : run in the background (detached mode)

-p hostPort:docker container port

--name give the name of your container

docker/getting-started (is the name of your image)

```
> docker ps
```

(you can see the container is up)

if you want to delete this container but if it is in running state you can't remove it directly
stop the container first and then remove

```
> docker stop my-website
```

```
> docker rm my-website
```

Remove the container directly from running state you can use -f flag.

```
> docker run -d -p 8082:80 --name my-website docker/getting-started
```

```
> docker rm my-website -f
```

Let's use the image which is MySQL

```
> docker pull MySQL
```

Now let's MySQL in a container

```
> sudo docker run --name mysql-server -e MYSQL_ROOT_PASSWORD=12345678 -d mysql
```

```
> sudo docker ps (mysql-server is up)
```

***** Let's Create Docker file for HTML Project *****

```
create index.html
```

```
create Dockerfile:
```

```
FROM nginx:1.20-alpine
```

```
COPY index.html /usr/share/nginx/html
```

```
EXPOSE 80
```

Now build image: `docker build -t sample-website .`

Now Run Image in container: `docker run -d --name webapp -p 8080:80 sample-website`

see the containers: `docker ps`

If its up check in browser: `localhost:8080/`

(this is how you can public your application in browser without setting any configuration)

