

Microservice orchestration platforms using Kubernetes

Docker basics

Konrad Dębiński

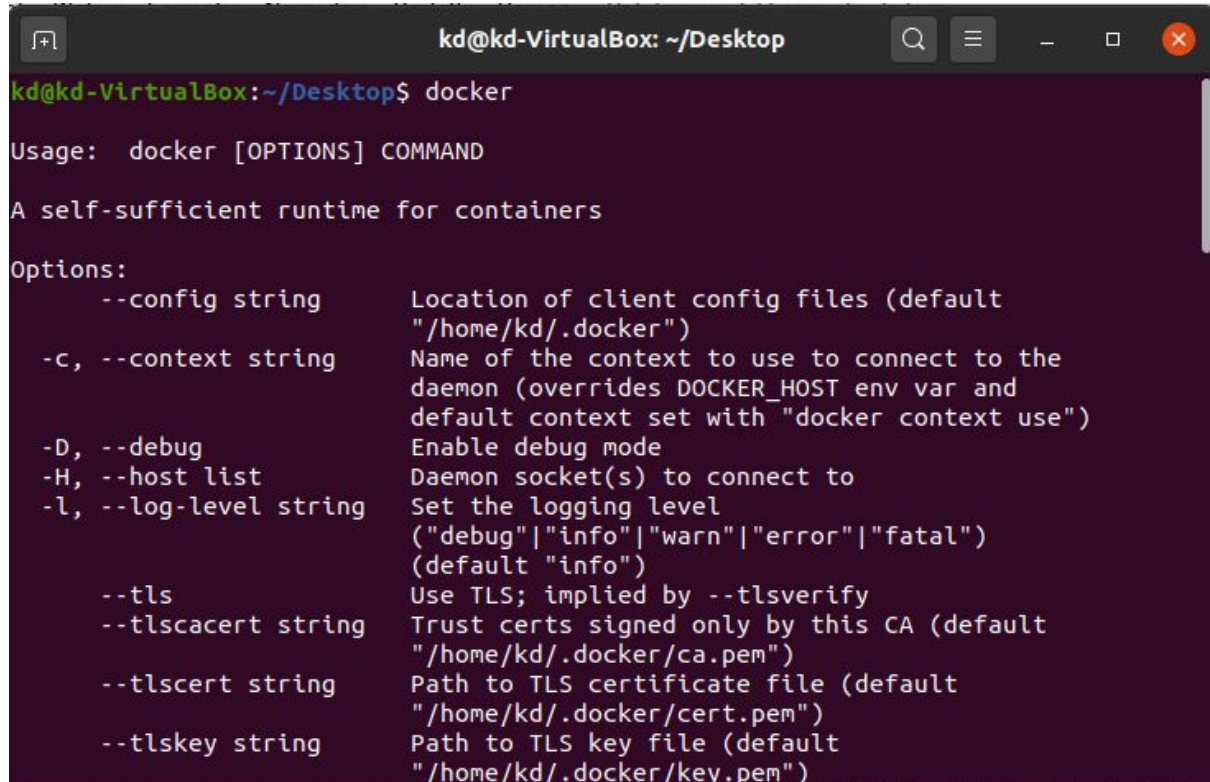
Introduction

During this laboratory our task was to deploy 5 different images on docker platform, to gain some basic knowledge about docker and its commands.

First of all the docker was installed with use of following commands:

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

It was correctly installed as docker command is now recognized by my system:

A terminal window titled 'kd@kd-VirtualBox: ~/Desktop' showing the output of the 'docker' command. The output includes the usage 'Usage: docker [OPTIONS] COMMAND', a description 'A self-sufficient runtime for containers', and a list of options with their descriptions. The options listed are: --config string (Location of client config files), -c, --context string (Name of the context to use to connect to the daemon), -D, --debug (Enable debug mode), -H, --host list (Daemon socket(s) to connect to), -l, --log-level string (Set the logging level), --tls (Use TLS), --tlscacert string (Trust certs signed only by this CA), --tlscert string (Path to TLS certificate file), and --tlskey string (Path to TLS key file).

```
kd@kd-VirtualBox:~/Desktop$ docker
Usage:  docker [OPTIONS] COMMAND

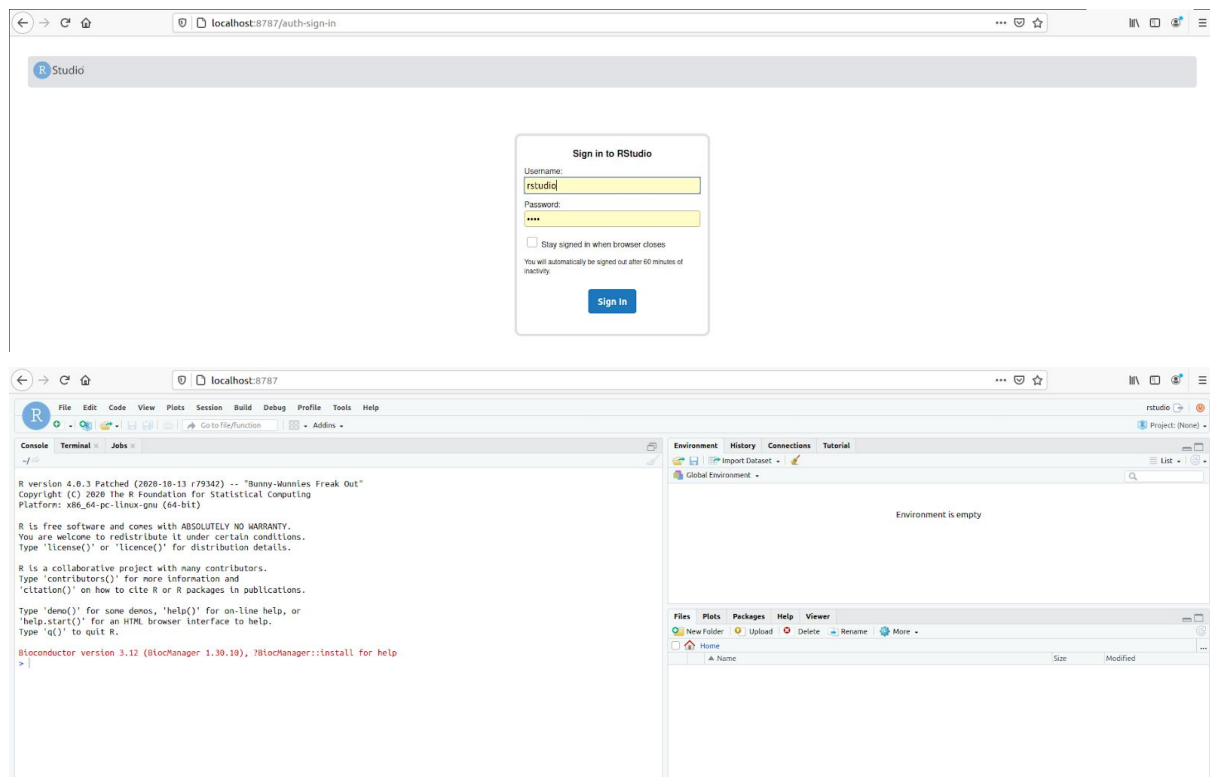
A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "/home/kd/.docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default
                        "/home/kd/.docker/ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "/home/kd/.docker/cert.pem")
  --tlskey string       Path to TLS key file (default
                        "/home/kd/.docker/key.pem")
```

As instructed by the tutor bioconductor container was deployed on the docker with usage of following command:

```
docker run \
  -e PASSWORD=bioc \
  -p 8787:8787 \
  bioconductor/bioconductor_docker:devel
```

It could be accessed on the <http://localhost:8787> address as shown in figure below:



After that I just found couple of other container images on docker hub and deployed them. With following command:

`docker pull image name`

List of all containers with use of command `docker ps -a` is presented below:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
afa76120c78d	solr	"docker-entrypoint.s..."	28 seconds ago	Up 26 seconds	0.0.0.0:8983->8983/tcp	mystifying_newto
460a82e3d123	jenkins	"/bin/tini -- /usr/l..."	3 minutes ago	Up 2 minutes	0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp	priceless_bartik
93033fa6465b	hello-world	"/hello"	14 minutes ago	Exited (0) 14 minutes ago		unruffled_mclare
01ce7840aed0	bioconductor/bioconductor_docker:devel	"/init"	28 minutes ago	Exited (0) 22 minutes ago		ecstatic_wright
c6cda32e66a9	gcr.io/k8s-minikube/kicbase:v0.0.13	"/usr/local/bin/entr..."	6 days ago	Exited (130) 28 minutes ago		minikube

As we can see two containers are still running, to stop them we can use following command:

`docker stop container id`

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
afa76120c78d	solr	"docker-entrypoint.s..."	3 minutes ago	Exited (137) 1 second ago
460a82e3d123	jenkins	"/bin/tini -- /usr/l..."	5 minutes ago	Up 5 minutes
93033fa6465b	hello-world	"/hello"	17 minutes ago	Exited (0) 17 minutes ago
01ce7840aed0	bioconductor/bioconductor_docker:devel	"/init"	30 minutes ago	Exited (0) 25 minutes ago
c6cda32e66a9	gcr.io/k8s-minikube/kicbase:v0.0.13	"/usr/local/bin/entr..."	6 days ago	Exited (130) 30 minutes ago

Conclusions

During this laboratory I gained knowledge about basics devoted to docker, how to pull an image, deploy and run it as well as stop it.