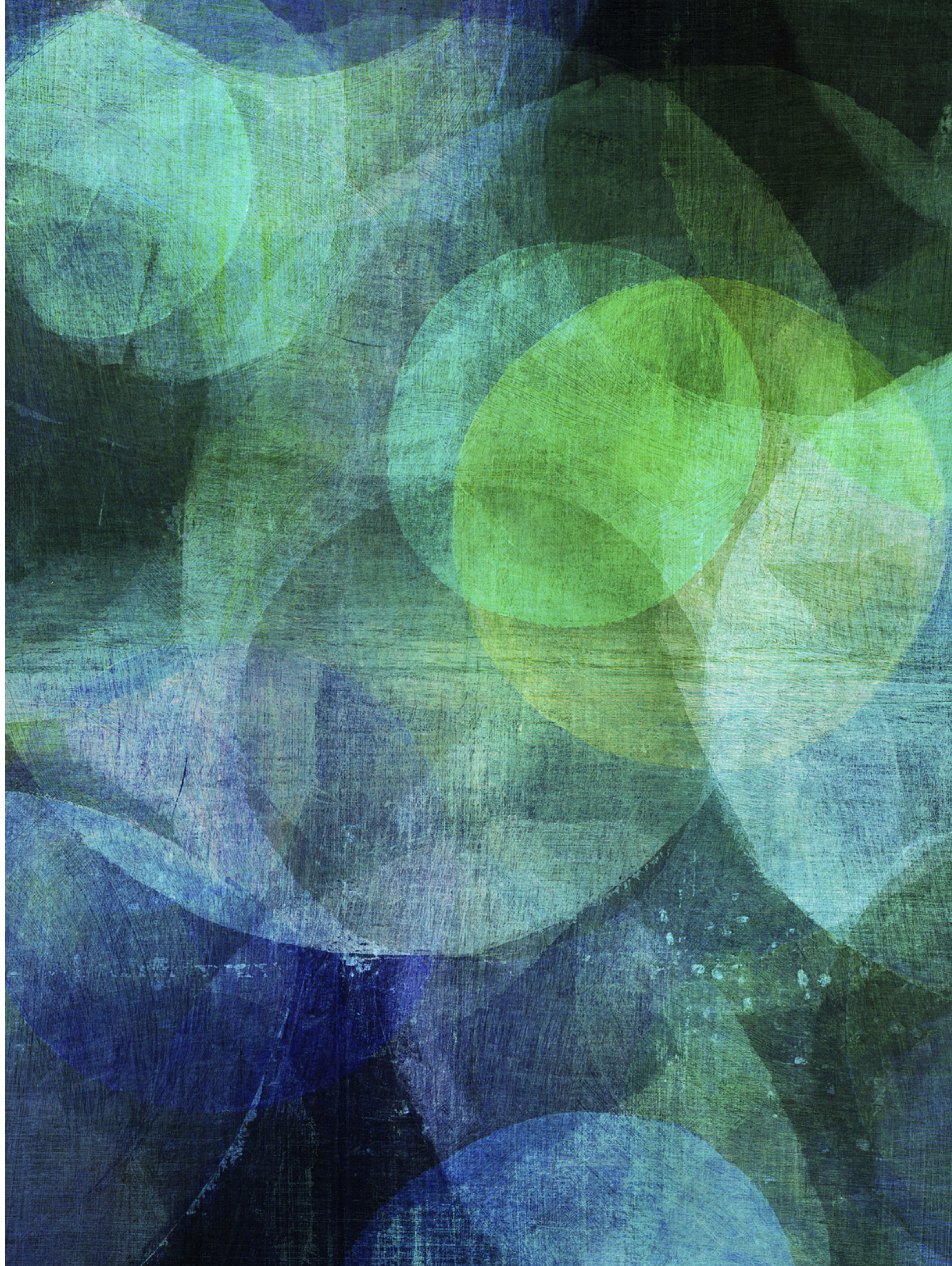
INTRODUCTION TO DOCKER

INTRODUCTION TO DOCKER

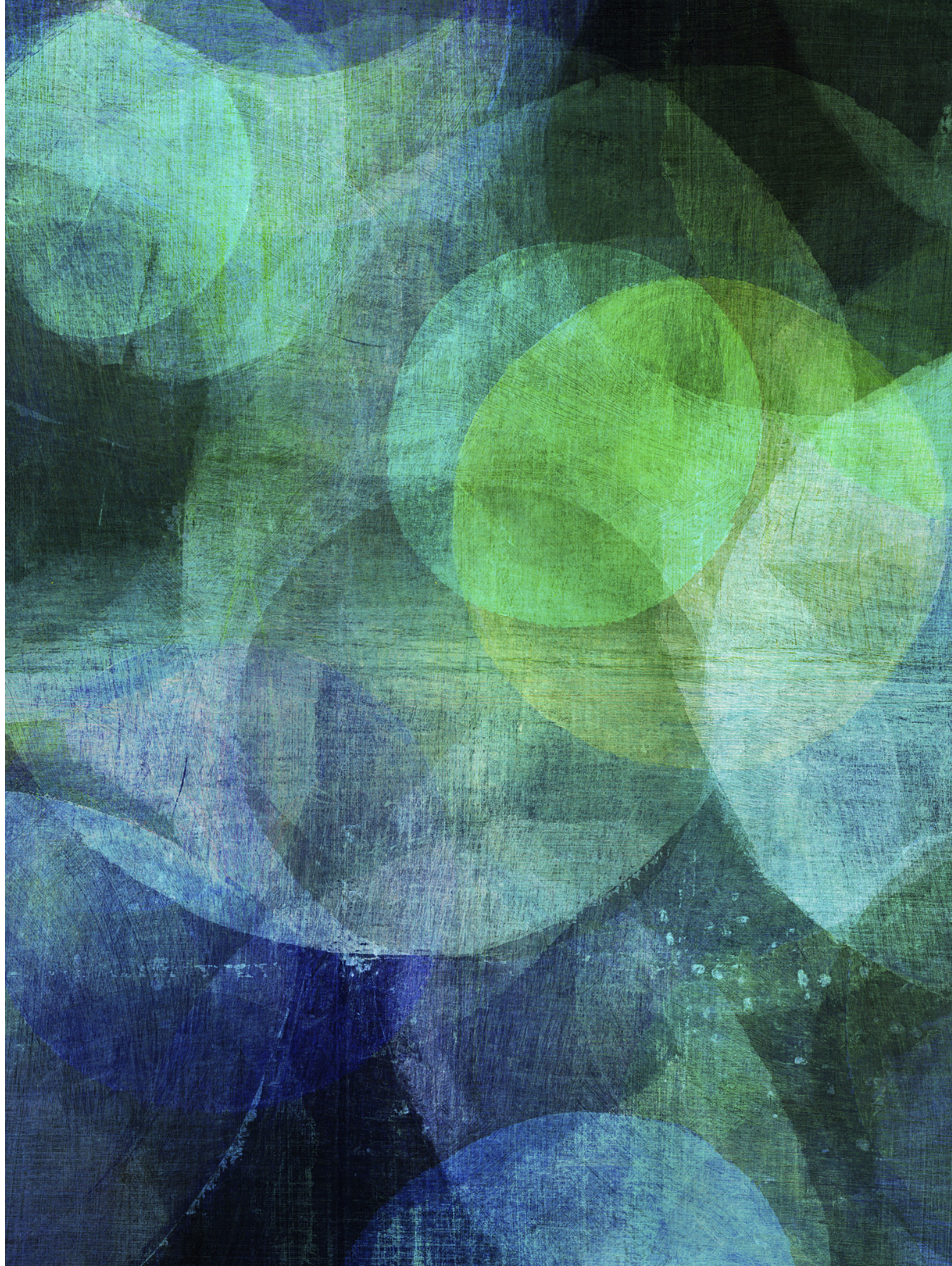
Docker Run and Docker Exec

AGENDA



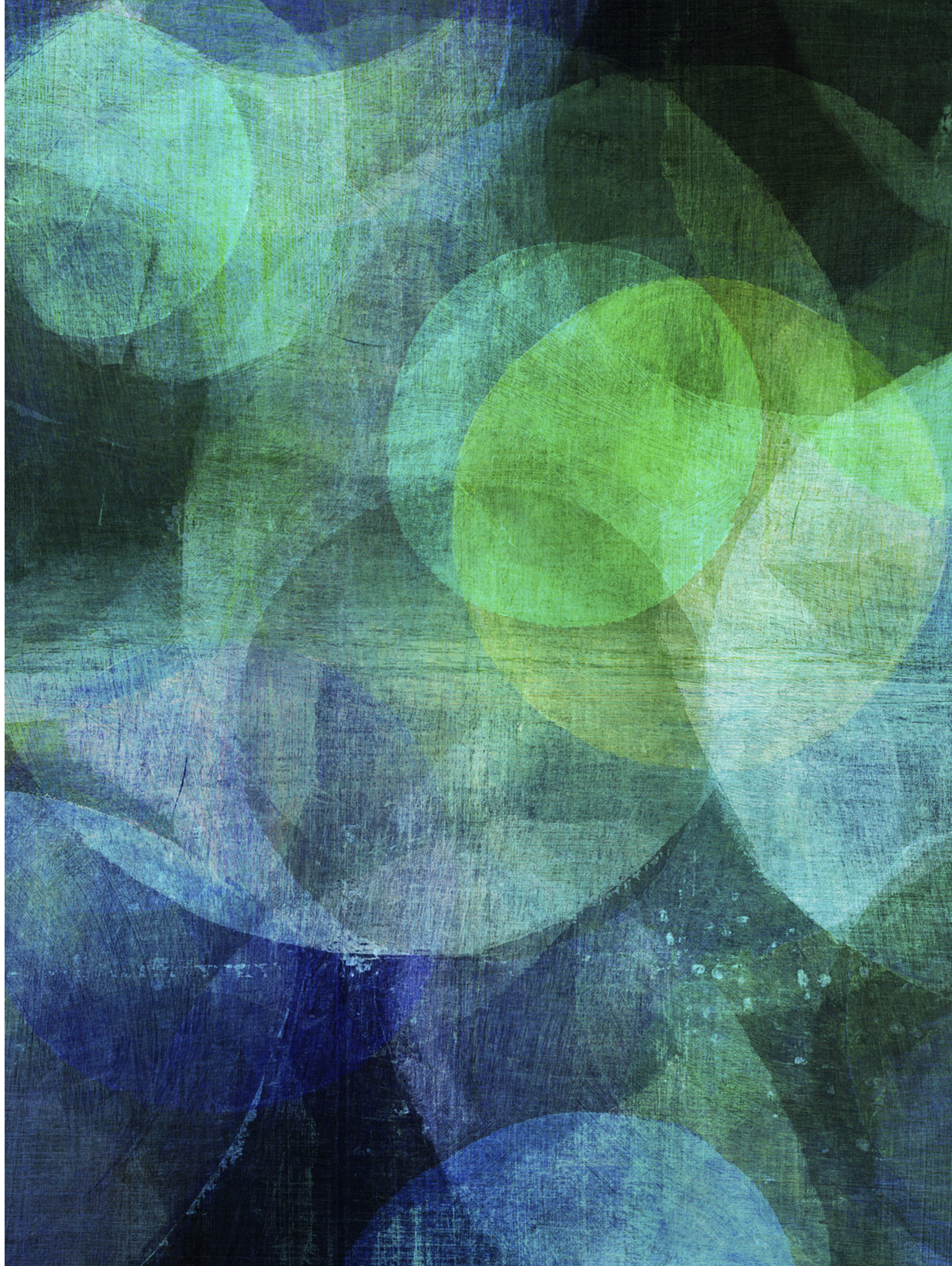
AGENDA

- Build Containers from Image



AGENDA

- Build Containers from Image
- Docker Run

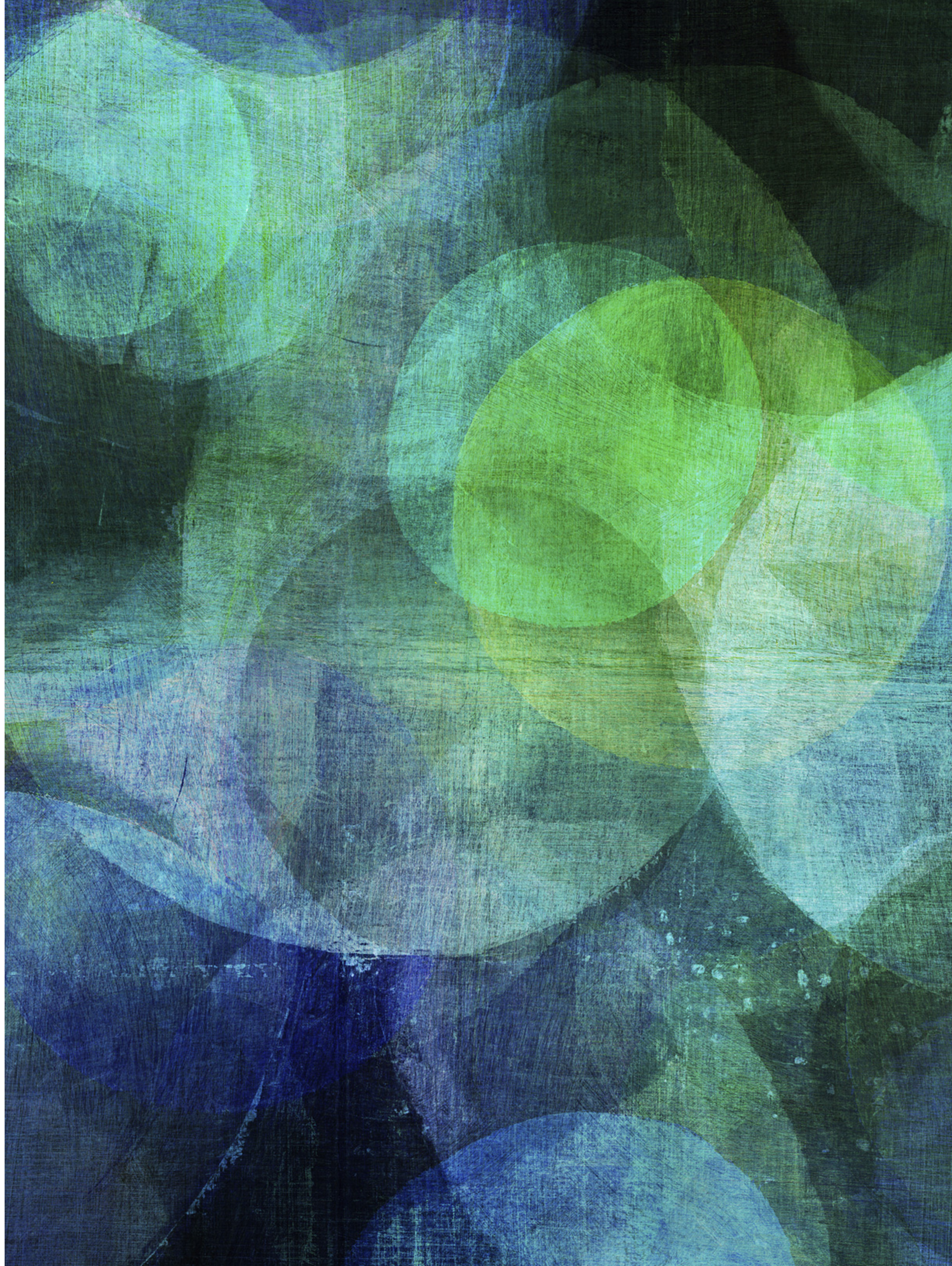


AGENDA

- Build Containers from Image

- Docker Run

- Assign Name to Container



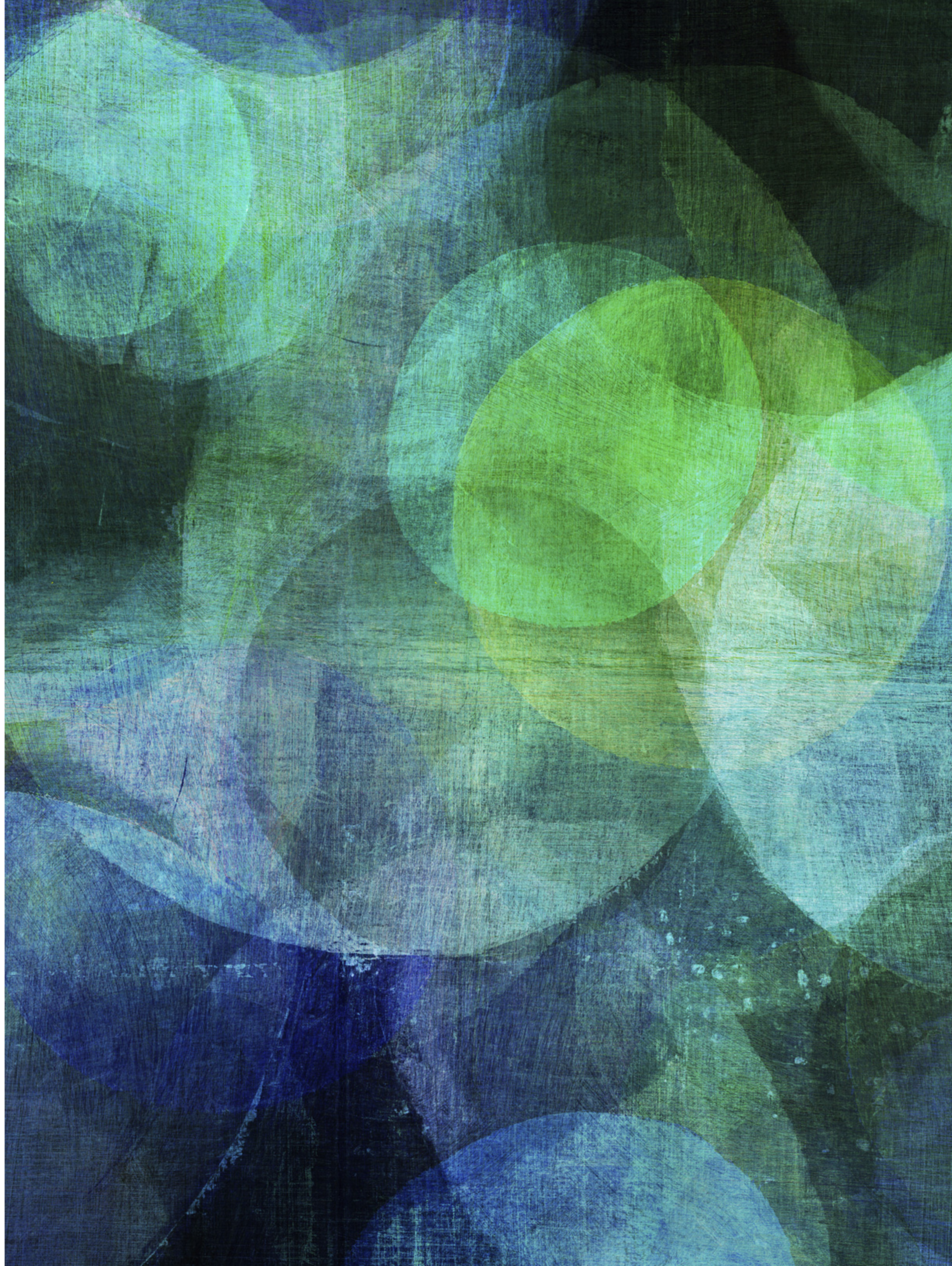
AGENDA

- Build Containers from Image

- Docker Run

- Assign Name to Container

- Environment Variables



AGENDA

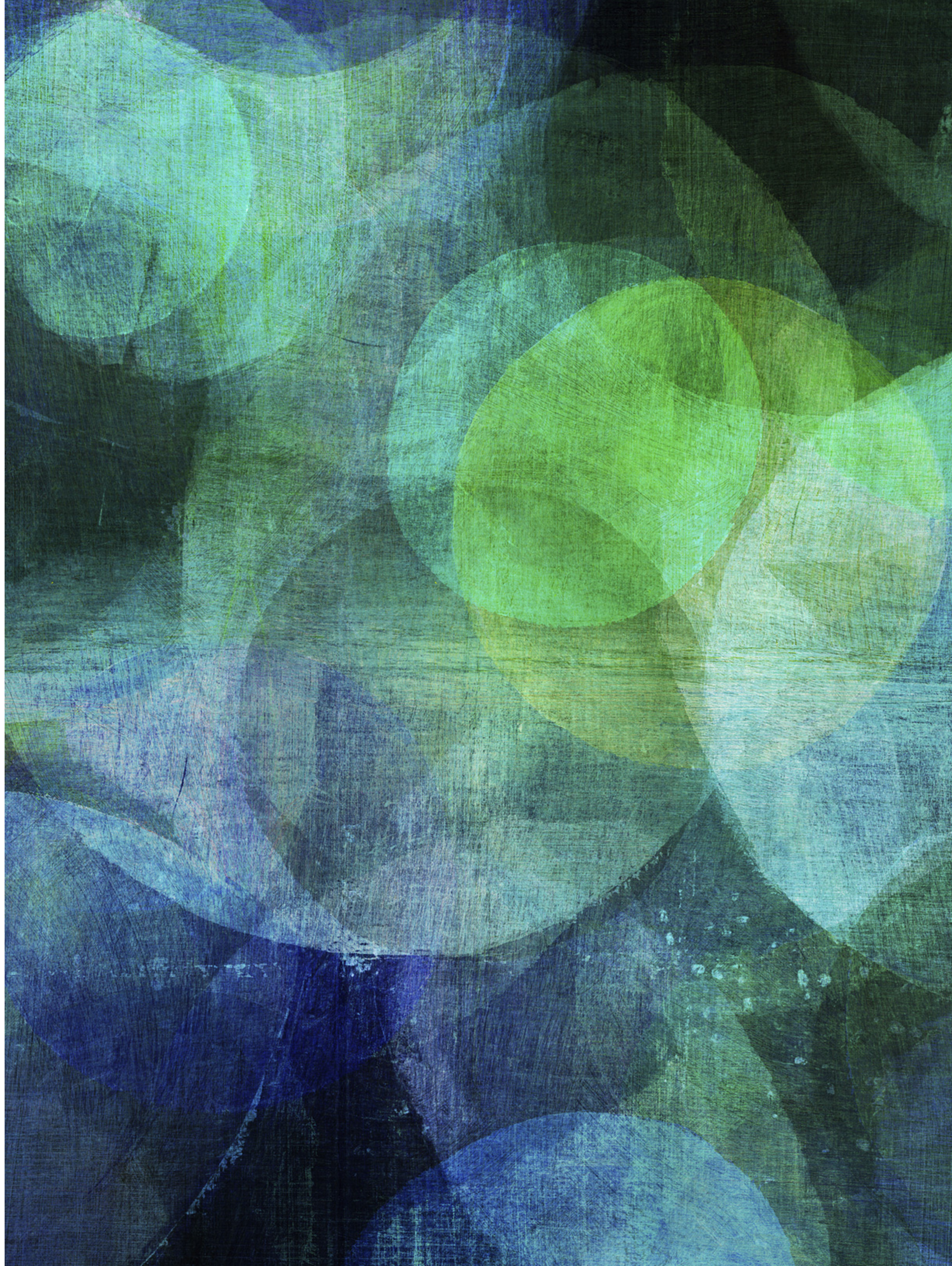
- Build Containers from Image

- Docker Run

- Assign Name to Container

- Environment Variables

- Run Docker in Background

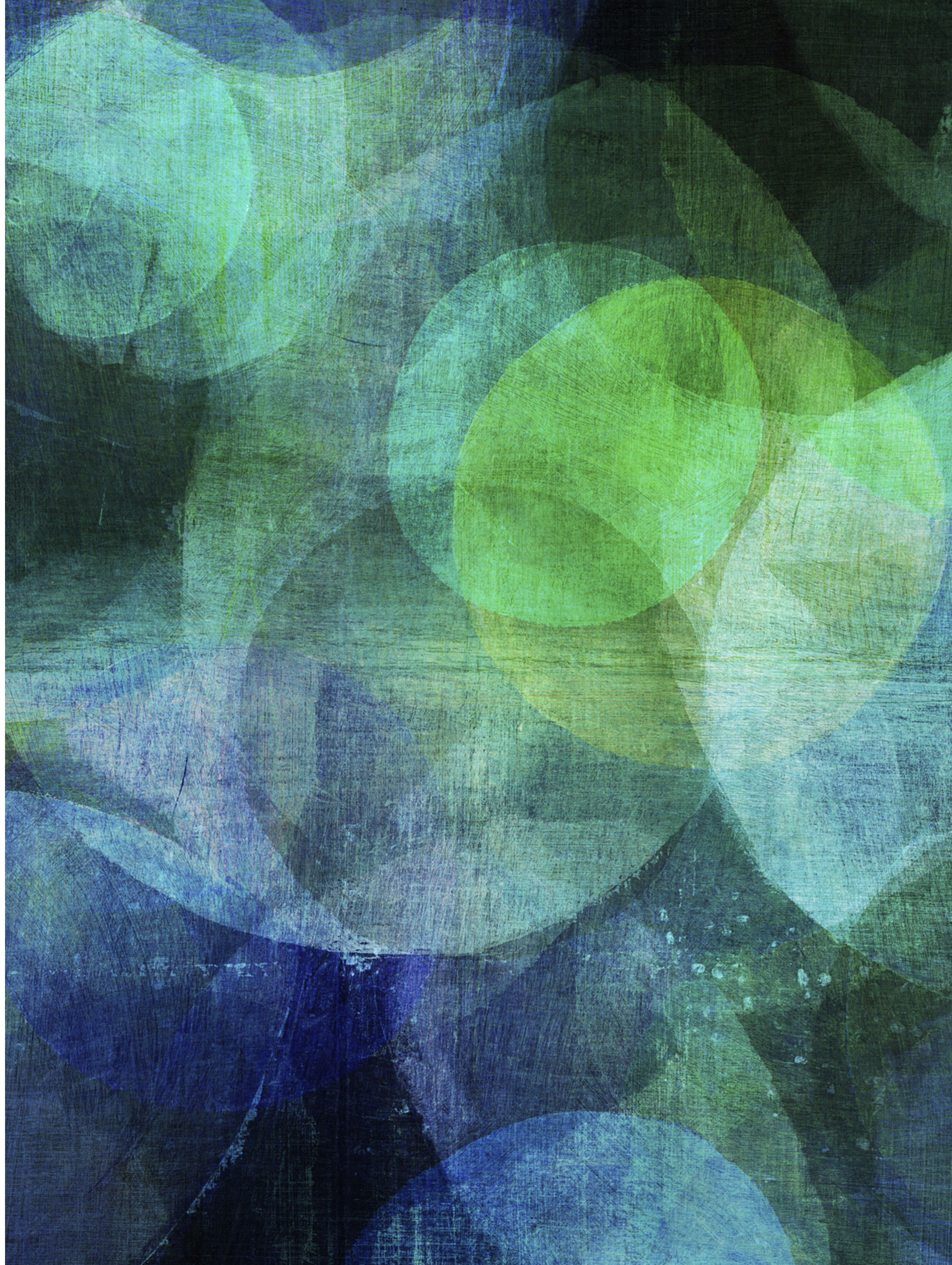


AGENDA

- Build Containers from Image

- Docker Run

- Assign Name to Container
- Environment Variables
- Run Docker in Background
- Auto Remove when exits

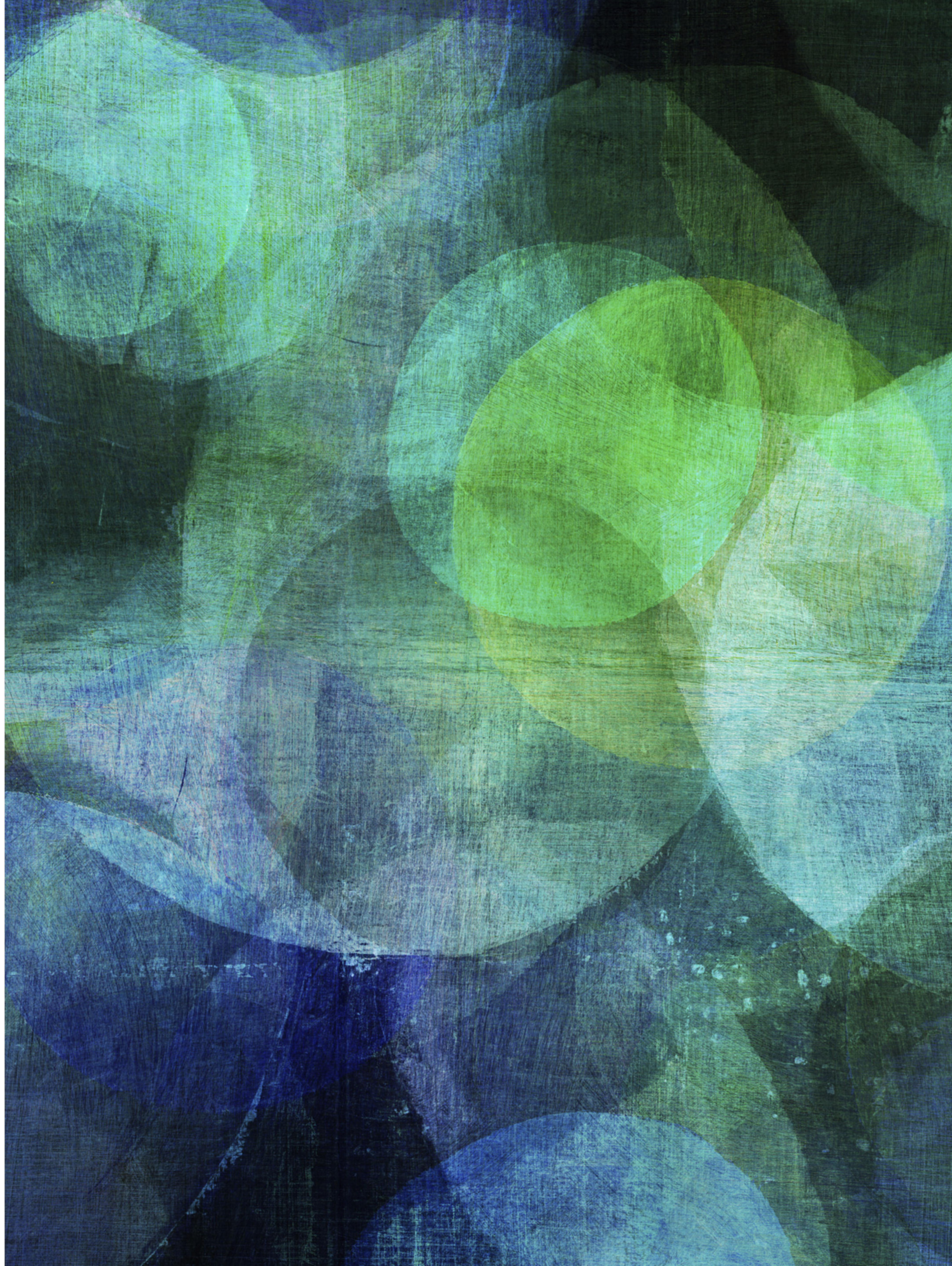


AGENDA

- Build Containers from Image

- Docker Run

- Assign Name to Container
- Environment Variables
- Run Docker in Background
- Auto Remove when exits
- Restart Policy



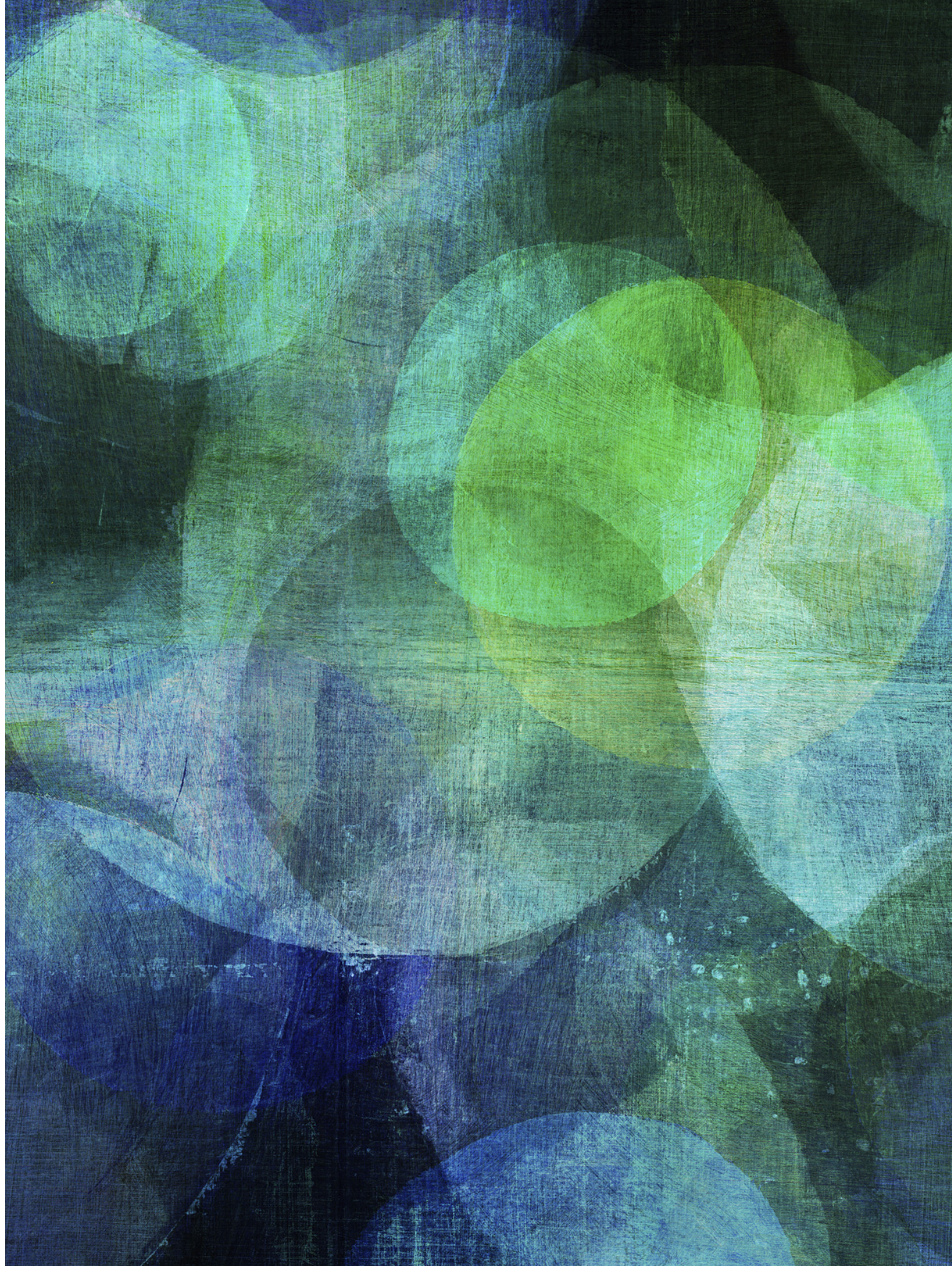
AGENDA

- Build Containers from Image

- Docker Run

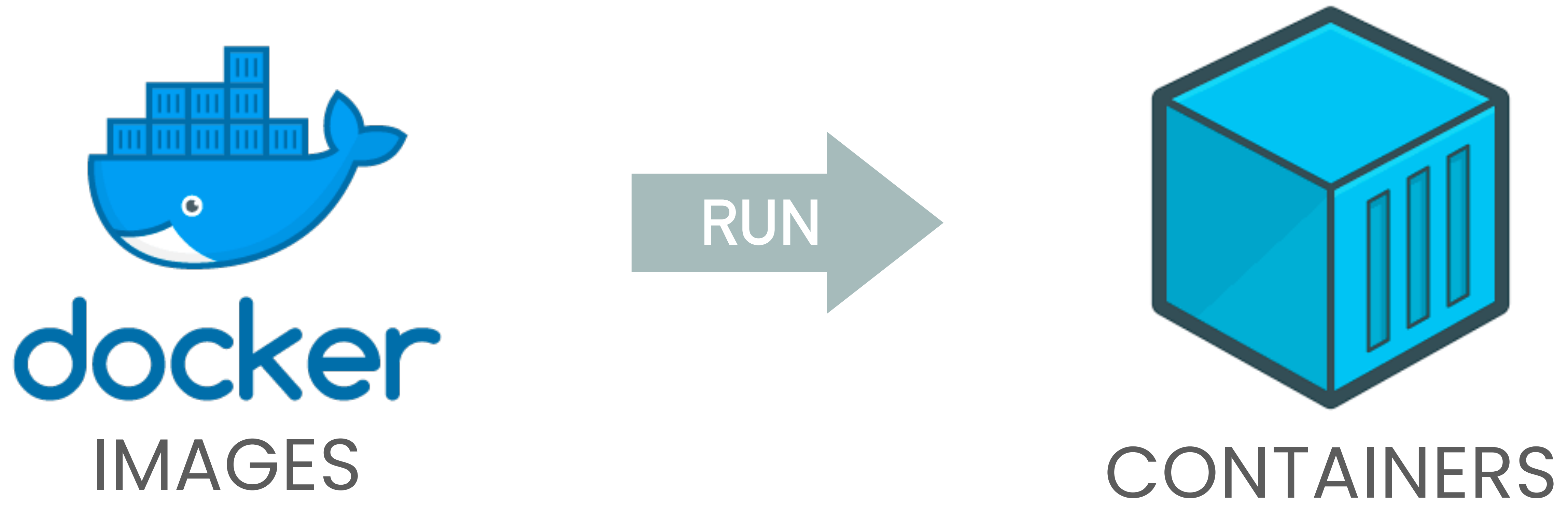
- Assign Name to Container
- Environment Variables
- Run Docker in Background
- Auto Remove when exits
- Restart Policy

- Docker EXEC Command



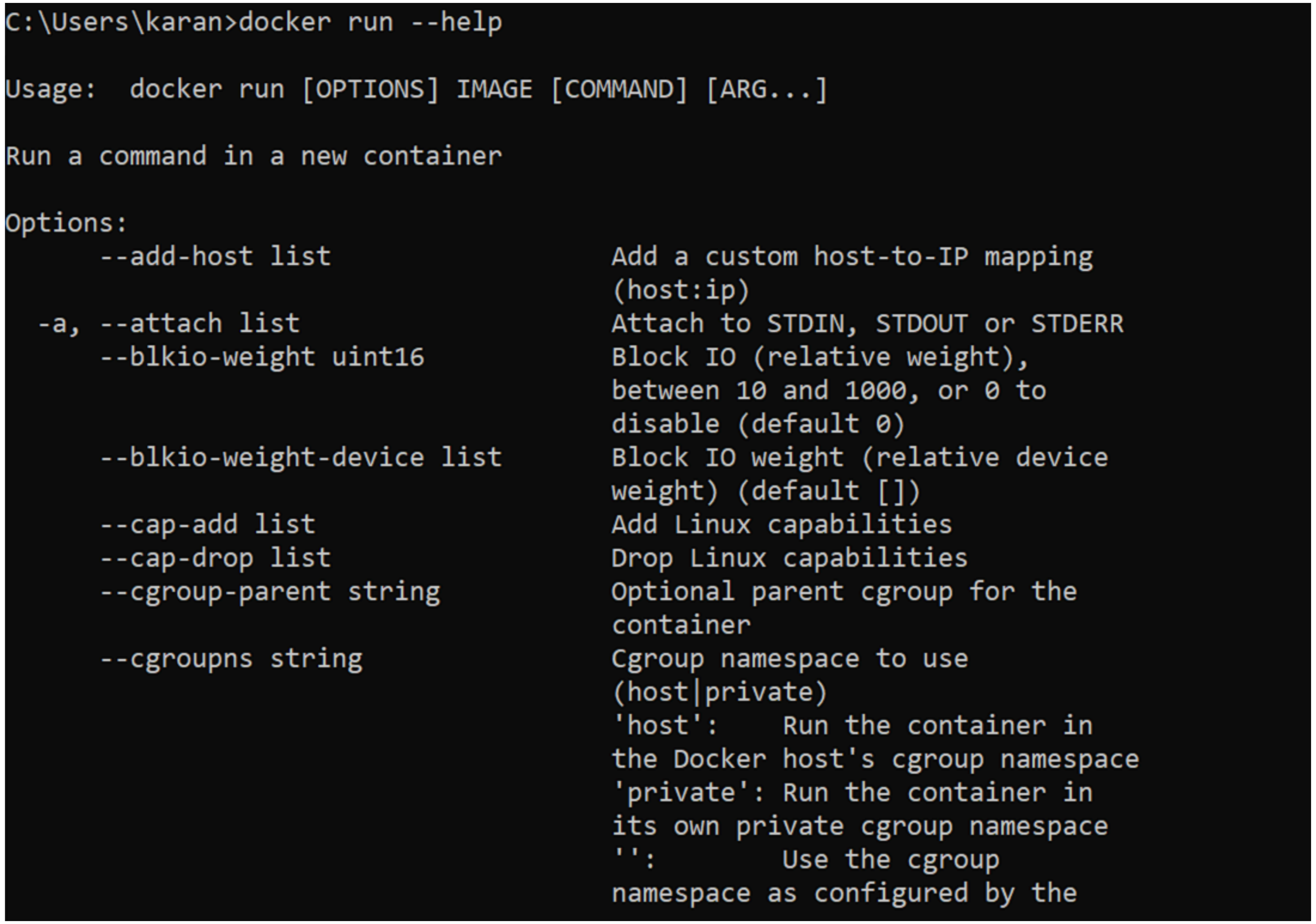
DOCKER RUN COMMAND

DOCKER RUN COMMAND



```
$ docker run hello-world

$ docker container run hello-world
```

DOCKER RUN COMMAND



```
C:\Users\karan>docker run --help

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container

Options:
      --add-host list                  Add a custom host-to-IP mapping
                                       (host:ip)
  -a, --attach list                    Attach to STDIN, STDOUT or STDERR
      --blkio-weight uint16            Block IO (relative weight),
                                       between 10 and 1000, or 0 to
                                       disable (default 0)
      --blkio-weight-device list       Block IO weight (relative device
                                       weight) (default [])
      --cap-add list                   Add Linux capabilities
      --cap-drop list                  Drop Linux capabilities
      --cgroup-parent string           Optional parent cgroup for the
                                       container
      --cgroupns string                Cgroup namespace to use
                                       (host|private)
                                       'host':    Run the container in
                                       the Docker host's cgroup namespace
                                       'private': Run the container in
                                       its own private cgroup namespace
                                       '':        Use the cgroup
                                       namespace as configured by the
```

RUN WITH SPECIFIC VERSION OF IMAGE (:)

RUN WITH SPECIFIC VERSION OF IMAGE (:)

```
$ docker run hello-world:linux
```

RUN WITH SPECIFIC VERSION OF IMAGE (:)

```
$ docker run hello-world:linux
```

RUN WITH SPECIFIC VERSION OF IMAGE (:)

```
$ docker run hello-world:linux
```

```
Unable to find image 'hello-world:linux' locally
linux: Pulling from library/hello-world
Digest: sha256:19c35675aac535e0f5803f12000ed7ffae510a43f1e3a839e7f4a9942a03dace
Status: Downloaded newer image for hello-world:linux

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
```

RUN WITH SPECIFIC VERSION OF IMAGE (:)

RUN WITH SPECIFIC VERSION OF IMAGE (:)

```
$ docker run hello-world:linux
```

RUN WITH SPECIFIC VERSION OF IMAGE (:)

```
$ docker run hello-world:linux
```

RUN WITH SPECIFIC VERSION OF IMAGE (:)

```
$ docker run hello-world:linux
```

```
$ docker image ls

REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    feb5d9fea6a5   4 months ago   13.3kB
hello-world   linux     feb5d9fea6a5   4 months ago   13.3kB
```

ASSIGN NAME TO CONTAINER (-- NAME)

ASSIGN NAME TO CONTAINER (-- NAME)

```
$ docker run --name my-name hello-world
```

ASSIGN NAME TO CONTAINER (-- NAME)

```
$ docker run --name my-name hello-world
```

ASSIGN NAME TO CONTAINER (-- NAME)

```
$ docker run --name my-name hello-world
```

```
$ docker container ls -a
CONTAINER ID   IMAGE         COMMAND    CREATED         STATUS                      PORTS     NAMES
07aa989fa98b   hello-world   "/hello"   6 seconds ago   Exited (0) 4 seconds ago              my-name
```

ENVIRONMENT VARIABLES (-E, -- ENV, —ENV-FILE)

ENVIRONMENT VARIABLES (-E, -- ENV, —ENV-FILE)

```
$ docker run -e MYVAR1=foo ubuntu:22.04
```

ENVIRONMENT VARIABLES (-E, -- ENV, —ENV-FILE)

```
$ docker run -e MYVAR1=foo ubuntu:22.04
```

=

```
$ docker run --env MYVAR2=bar ubuntu:22.04
```

ENVIRONMENT VARIABLES (-E, -- ENV, —ENV-FILE)

```
$ docker run -e MYVAR1=foo ubuntu:22.04
```

=

```
$ docker run --env MYVAR2=bar ubuntu:22.04
```

=

```
$ docker run --env-file ./env.txt ubuntu:22.04
```

RUN WITH STDIN AND INTERACTIVE MODE (-IT)

RUN WITH STDIN AND INTERACTIVE MODE (-IT)

```
$ docker run -it ubuntu bash
```

RUN WITH STDIN AND INTERACTIVE MODE (-IT)

```
$ docker run -it ubuntu bash
```

RUN WITH STDIN AND INTERACTIVE MODE (-IT)

```
$ docker run -it ubuntu bash
```

```
root@afcf2ad1d183:/# ls -lt
total 48
drwxr-xr-x   5 root root  360 Jan 29 17:11 dev
dr-xr-xr-x 175 root root    0 Jan 29 17:11 proc
dr-xr-xr-x  11 root root    0 Jan 29 17:11 sys
drwxr-xr-x   1 root root 4096 Jan 29 17:11 etc
drwxr-xr-x   5 root root 4096 Jan  5 16:50 run
drwxrwxrwt   2 root root 4096 Jan  5 16:50 tmp
drwx------   2 root root 4096 Jan  5 16:50 root
drwxr-xr-x  11 root root 4096 Jan  5 16:50 var
drwxr-xr-x   2 root root 4096 Jan  5 16:47 media
drwxr-xr-x   2 root root 4096 Jan  5 16:47 mnt
drwxr-xr-x   2 root root 4096 Jan  5 16:47 opt
drwxr-xr-x   2 root root 4096 Jan  5 16:47 srv
drwxr-xr-x  13 root root 4096 Jan  5 16:47 usr
```

REMOVE WHEN EXITS (-- RM)

REMOVE WHEN EXITS (-- RM)

```
$ docker run --rm --name my-name hello-world
```

REMOVE WHEN EXITS (-- RM)

```
$ docker run --rm --name my-name hello-world
```

REMOVE WHEN EXITS (-- RM)

```
$ docker run --rm --name my-name hello-world
```

```
$ docker container ls -a
CONTAINER ID    IMAGE           COMMAND      CREATED          STATUS                             PORTS       NAMES
```

REMOVE ALL STOPPED CONTAINERS

REMOVE ALL STOPPED CONTAINERS

```
$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
c2d836e6e39edad61b057e6d41407d92b179c8fae48e9e1e9ffe106f87ff1c0b

Total reclaimed space: 1.093kB
```

REMOVE ALL STOPPED CONTAINERS

```
$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
c2d836e6e39edad61b057e6d41407d92b179c8fae48e9e1e9ffe106f87ff1c0b

Total reclaimed space: 1.093kB
```

REMOVE ALL STOPPED CONTAINERS

```
$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
c2d836e6e39edad61b057e6d41407d92b179c8fae48e9e1e9ffe106f87ff1c0b

Total reclaimed space: 1.093kB
```

```
$ docker container ls -a
CONTAINER ID   IMAGE         COMMAND    CREATED         STATUS                     PORTS     NAMES
```

RUN CONTAINER AS A SERVICE/SERVER (LONG RUNNING PROCESS)

RUN CONTAINER AS A SERVICE/SERVER (LONG RUNNING PROCESS)

```
$ docker run --name my-web nginx:1.20.2
```

RUN CONTAINER AS A SERVICE/SERVER (LONG RUNNING PROCESS)

```
$ docker run --name my-web nginx:1.20.2
```

RUN CONTAINER AS A SERVICE/SERVER (LONG RUNNING PROCESS)

```
$ docker run --name my-web nginx:1.20.2
```

```
Unable to find image 'nginx:1.20.2' locally
1.20.2: Pulling from library/nginx
5eb5b503b376: Pull complete
cdfeb356c029: Pull complete
d86da7454448: Pull complete
7976249980ef: Pull complete
8f66aa6726b2: Pull complete
c004cabebe76: Pull complete
Digest: sha256:02923d65cde08a49380ab3f3dd2f8f90aa51fa2bd358bd85f89345848f6e6623
Status: Downloaded newer image for nginx:1.20.2
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
```

RUN CONTAINER IN BACKGROUND (-- DATACH, -D)

RUN CONTAINER IN BACKGROUND (-- DATACH, -D)

```
$ docker run --detach --name my-web nginx:1.20.2
44ae8b7d1bc01a57547190007890fa2b03b0ac5a5ddd1a2568f88e7ea3f19596
```

RUN CONTAINER IN BACKGROUND (-- DATACH, -D)

```
$ docker run --detach --name my-web nginx:1.20.2
44ae8b7d1bc01a57547190007890fa2b03b0ac5a5ddd1a2568f88e7ea3f19596
```

RUN CONTAINER IN BACKGROUND (-- DATACH, -D)

```
$ docker run --detach --name my-web nginx:1.20.2
44ae8b7d1bc01a57547190007890fa2b03b0ac5a5ddd1a2568f88e7ea3f19596
```

```
$ docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS         PORTS     NAMES
44ae8b7d1bc0   nginx:1.20.2   "/docker-entrypoint.…"   8 minutes ago    Up 8 minutes   80/tcp    my-web
```

RESTART POLICY (-- RESTART)

A restart policy controls whether the Docker daemon restarts a container after exit. Docker supports 4 restart policies

- **no** Do not automatically restart the container when it exits. This is the default.
- **on-failure[:max-retries]** Restart only if the container exits with a non-zero exit status. Optionally, limit the number of restart retries the Docker daemon attempts.
- **unless-stopped** Restart the container unless it is explicitly stopped or Docker itself is stopped or restarted.
- **always** Always restart the container regardless of the exit status. When you specify always, the Docker daemon will try to restart the container indefinitely. The container will also always start on daemon startup, regardless of the current state of the container.

RESTART POLICY (-- RESTART)

RESTART POLICY (-- RESTART)

```
$ docker run --restart always hello-world
```

RESTART POLICY (-- RESTART)

```
$ docker run --restart always hello-world
```

RESTART POLICY (-- RESTART)

```
$ docker run --restart always hello-world
```

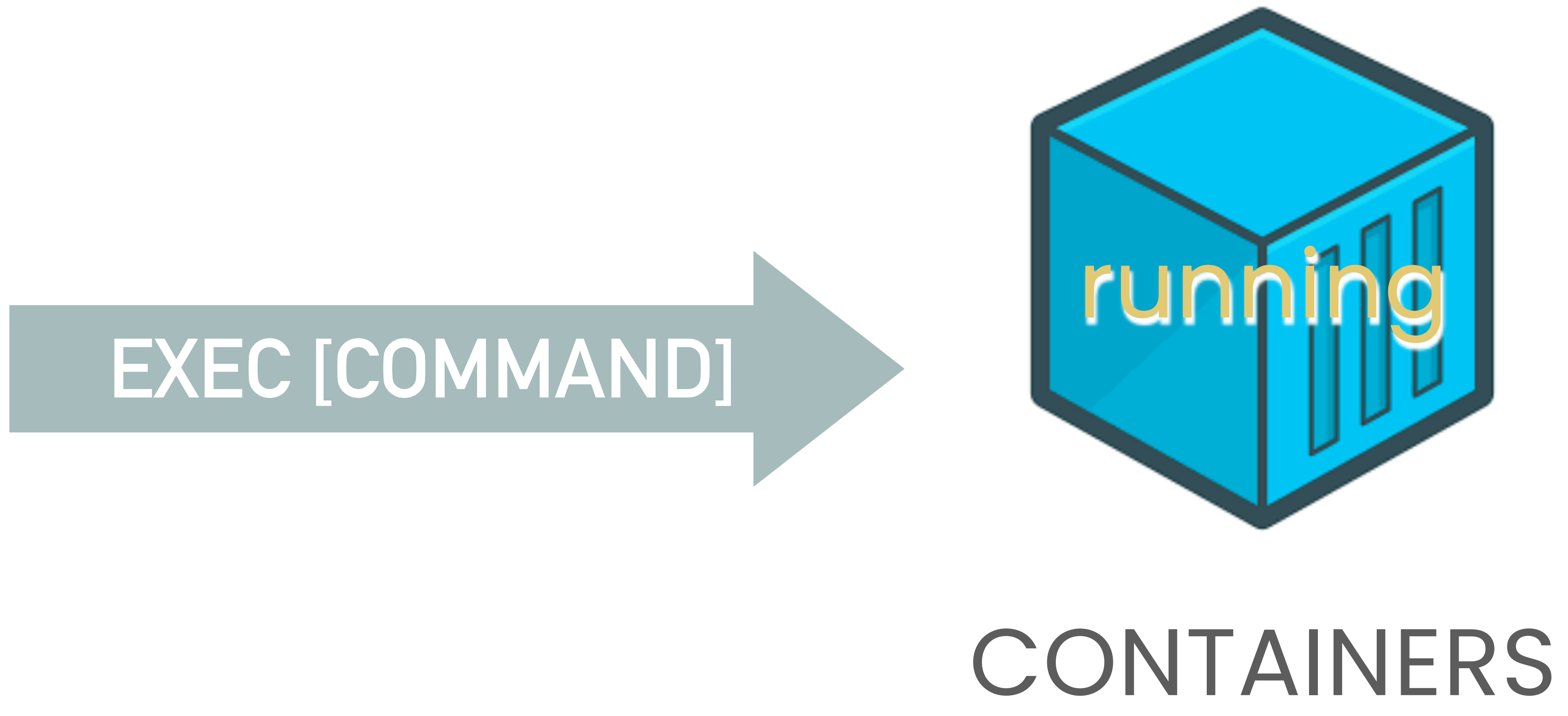
```
$ docker container ls -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                          PORTS     NAMES
a781df870a94   hello-world   "/hello"   15 seconds ago   Restarting (0) 2 seconds ago              unruffled_easley

$ docker container ls -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                           PORTS     NAMES
a781df870a94   hello-world   "/hello"   32 seconds ago   Restarting (0) 12 seconds ago              unruffled_easley

$ docker container ls -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                          PORTS     NAMES
a781df870a94   hello-world   "/hello"   36 seconds ago   Restarting (0) 1 second ago               unruffled_easley
```

DOCKER EXEC COMMAND

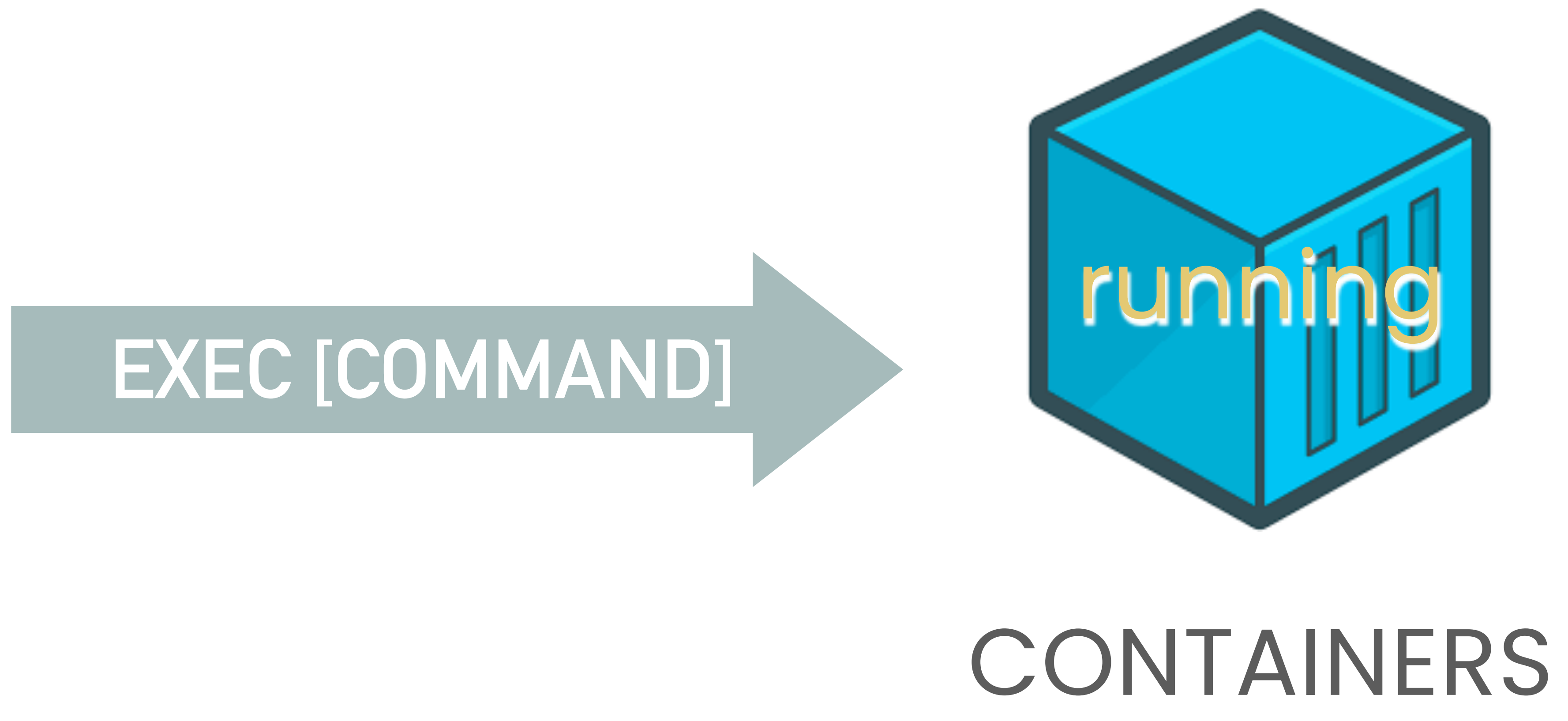
DOCKER EXEC COMMAND

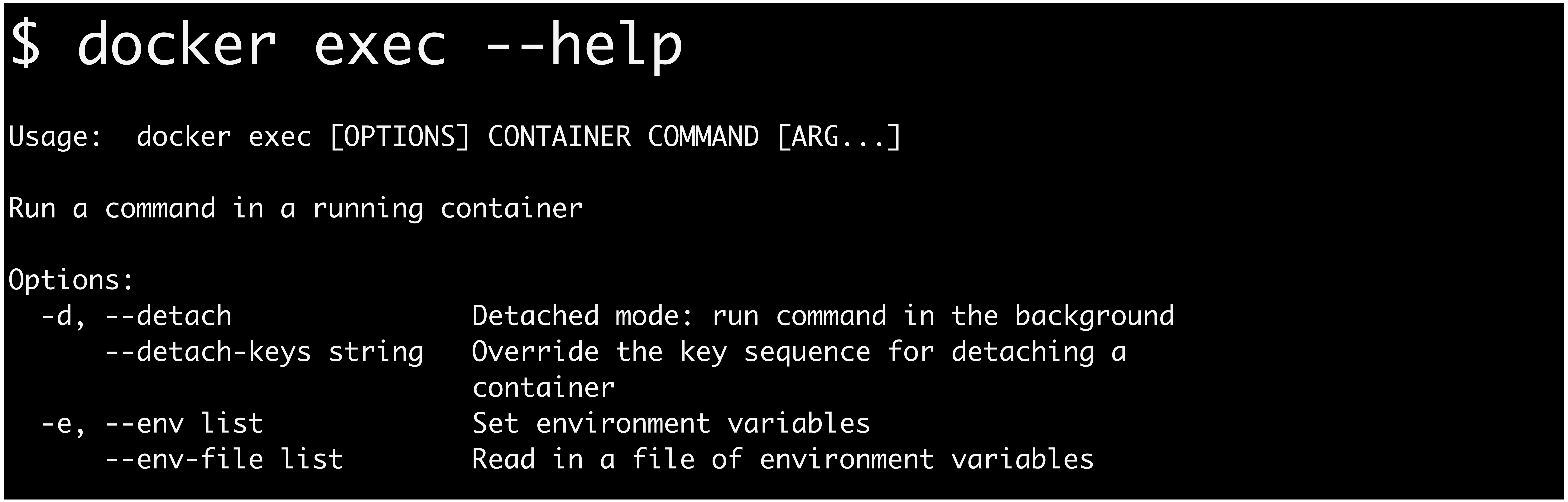
CONTAINERS

DOCKER EXEC COMMAND



DOCKER EXEC COMMAND





```
$ docker exec --help

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container

Options:
  -d, --detach               Detached mode: run command in the background
      --detach-keys string   Override the key sequence for detaching a
                             container
  -e, --env list             Set environment variables
      --env-file list        Read in a file of environment variables
```

DOCKER EXEC COMMAND

```
$ docker run ––name my-linux –it ubuntu bash
root@afcf2ad1d183:/#
```

DOCKER EXEC COMMAND

```
$ docker run ––name my-linux –it ubuntu bash
root@afcf2ad1d183:/#
```

```
$ docker exec my-linux touch /tmp/execWorks
```

DOCKER EXEC COMMAND

```
$ docker run ––name my-linux –it ubuntu bash
root@afcf2ad1d183:/#
```

```
$ docker exec my-linux touch /tmp/execWorks
```

```
root@afcf2ad1d183:/# ls -lt /tmp
total 0
-rw-r--r-- 1 root root 0 Jan 29 17:22 execWorks

root@afcf2ad1d183:/#
```

LET TRY !

THANK YOU