

Environment variables

- This command starts a new container using the Alpine Linux image and sets the environment variable MY_ENV_VAR to the value hello_world. The env command at the end simply prints out all the environment variables set in the container, so we can see that MY_ENV_VAR is indeed set to hello_world

```
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-7$ docker run --rm -e MY_ENV_VAR=hello_world alpine env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=a60fc4db533a
MY_ENV_VAR=hello_world
HOME=/root
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-7$
```

- Setting environment variables is a powerful way to customize the configuration of Docker images, as many Docker images use them to drive their configuration. For example, the postgres image allows you to specify default user/password by using environment variables. You can also use environment variables to set sensitive configurations such as API keys and database URLs for your own custom images. Additionally, to connect to processes running on your Docker containers, you need to expose and map ports between Docker containers and your host machine.

```
ankit@sf-cpu-082:~/Documents/Assignment/Docker/docker-training/Task-7$ docker run --rm -e POSTGRES_USER=archer -e POSTGRES_PASSWORD=guest postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
26c5c85e47da: Already exists
1c30a4c3f519: Pull complete
d5c0f1ae682d: Pull complete
1b1b2890ec0f: Pull complete
391087799df7: Pull complete
b413b4057e31: Pull complete
4fa4edfeab8b: Pull complete
b0a4d596bc61: Pull complete
f6d73cd87199: Pull complete
62b0bb33c69b: Pull complete
bb0ddb7e7f1a: Pull complete
583ec94d38ee: Pull complete
efd2a922e82: Pull complete
Digest: sha256:6cc97262444f1c45171081bc5ald4c28b883ea46a6e0d1a45a8eac4a7f4767ab
Status: Downloaded newer image for postgres:latest
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /var/lib/postgresql/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Etc/UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

pg_ctl -D /var/lib/postgresql/data -l logfile start
```

- **Ports**
- In many cases, you'll want to connect to processes running on your Docker containers, such as a web server on port 80.
- By default, however, Docker containers are network inaccessible from your local machine.