So in order to pass arguments we need to use keyword **environment:**

For example:

**version: "3"**

**services:**

**squre:**

**image: dimbas/passing\_arguments\_to\_javaprogram\_using\_dickerfile:latest**

**environment:**

**- VARIABLE=54 23**

**docker-compose up** -> the command to run the file (we need to be in file`s directory)

So in this case the arguments hard coded in a file and in order to change it you need to change the file itself, theris no way to change it by using: docker-compose up command.

Это мне кажется не совсем по теме так как использует run

In order to be able to pass dynamic arguments to the yaml file.(any value that you want)

For example:

**version: "3"**

**services:**

**squre:**

**image: dimbas/passing\_arguments\_to\_javaprogram\_using\_dickerfile:latest**

**environment:**

**- VARIABLE** <-this is the name of variable we are using, and to this variable we should refer in a run command

**docker-compose run -e VARIABLE="25" squre**

-e <- is the key that tells the docker that we are gonna use dynamic variable

squre <- is the name of the service

—-----------------------—------------------------------------------ **env file** —-----------------------------------------------------

On windows we can use:

.env file, it should be in the same directory as an yaml file. an it should have structure key:value (no space).

For example:

yaml file:

**version: "3"**

**services:**

**squre:**

**image: dimbas/passing\_arguments\_to\_javaprogram\_using\_dickerfile:latest**

**environment:**

**VARIABLE: "${VARIABLE}"** <- this is first variable we need to pass value for

**alpine:**

**image: alpine**

**entrypoint: "wget ${KURLI}"** <- this is second variable we need to pass value for

**depends\_on:**

**- squre**

**working\_dir: /directory/**

**volumes:**

**- C:\forExperements\docker\_experements\YAML\passing arguments\:/directory/**

.env file:

**URL=**[**https://www.baeldung.com/ops/docker-compose**](https://www.baeldung.com/ops/docker-compose) **←** value for the first variable

**VARIABLE=10 20 ←**value for the second variable

to run in cmd: docker-compose up

—----------------—----------------------------------------- **OS VARIABLES** —--------------------------------------------

we can set appropriate variables in OS (windows), for this use:

***set variable\_name=value***

To unset/delete variable use:

set variable\_name=

To see variable

set variable\_name

After variables are set, use:

docker-compose up

—---------------------------------------------------- **value by default** —-----------------------------------------

We can assign default value to variable, so if we do not pass any value with run instruction the programm will execute using value by default

**version: "3"**

**services:**

**squre:**

**image: dimbas/passing\_arguments\_to\_javaprogram\_using\_dickerfile:latest**

**environment:**

**VARIABLE: "${VARIABLE:-10}"**

**alpine:**

**image: alpine**

**entrypoint: "wget ${KURLI:-google.com}"**

**depends\_on:**

**- squre**

**working\_dir: /directory/**

**volumes:**

**- C:\forExperements\docker\_experements\YAML\passing arguments\:/directory/**

—--------—------------------------------------------------------ **mupltiple ENV** —------------------------------------------------------

We can have multiple env files with different values for variables, it is useful for example for different scenarios. In order to use them it should be in the same direction as the .yaml file.

**docker-compose up** ← execute the .env file by default (without name only extinction)

**docker-compose --env-file demo.env up**← syntax for executing specified env file

**--env-file** ←this is a keyword for docker to know that we are about to use the specific env file

**demo.env** ← this is the name of specific .env file to execute.