CLOUD COMPUTING

ASSIGNMENT 1

GOAL

- Setup environment for this and future assignments
- Write an HTTP server application in a Docker container
- Execute your application from container

PREREQUISITE

- Gitlab account
- Linux
- Docker
- Git

GITLAB SETUP

1. **On gitlab.com, create a repo named (exactly)** *cloudcomputing-2019-assignments*
2. **Add Rémi's account (***dulongr***) and grant him access to the repository with guest access rights (read-only)**

GIT

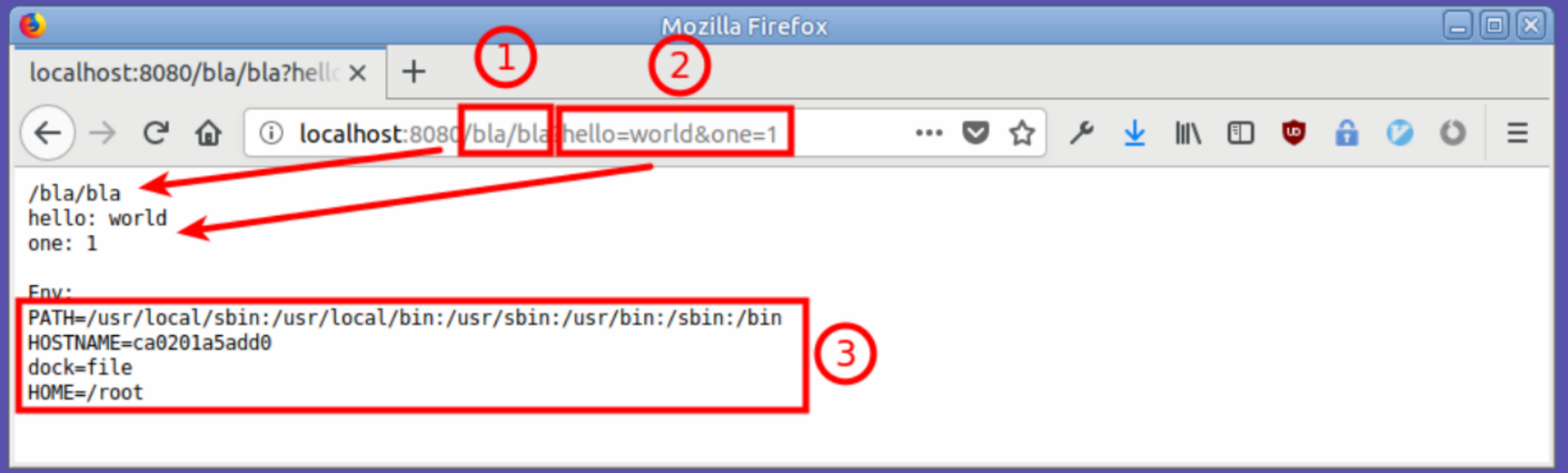
- Replace XXX by your gitlab's user

```
$ git clone git@gitlab.com:XXX/cloudcomputing-2019-assignments.git
Clonage dans 'cloudcomputing-2019-assignments'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 0), reused 0 (delta 0)
Réception d'objets: 100% (9/9), fait.
$ cd cloudcomputing-2019-assignments
$ mkdir assignment1
$ cd assignment1
```

- In this folder, prepare a Dockerfile and all other files required to run your app

OBJECTIVE

1. Display URL path
2. Extract & show URL parameters
3. Show environment variables



EXAMPLE: SERVER.GO 1/2

```
package main

import (
    "fmt"
    "net/http"
)

func main() {
    port := 8080
    fmt.Println("Listening on port: ", port)
    http.HandleFunc("/", handler)
    http.ListenAndServe(":8080", nil)
}
```

EXAMPLE: SERVER.GO 2/2

```
func handler(w http.ResponseWriter, r *http.Request) {
    keys, ok := r.URL.Query()["hello"]
    if !ok {
        fmt.Fprintf(w, "Param 'hello' is missing")
        return
    }
    key := keys[0]
    fmt.Fprintf(w, "Hello: " + key)
}
```

BUILD & RUN

```
$ sudo apt install golang
$ go build -o server .
```

```
$ ./server
Listening on port:  8080
```

EXAMPLE: DOCKERFILE

Dockerfile:

```
FROM debian
RUN apt update && apt -f -y install golang
ADD . /go/
WORKDIR /go
RUN go build -o server .
ENTRYPOINT ["/go/server"]
EXPOSE 8080
```

Build & run (from Docker):

```
$ docker build -t assignment1 .
$ docker run -p 8080:8080 assignment1
Listening on port:  8080
```

TEST MY SOLUTION

- Solution pushed on my Docker Hub account (you don't have to do that)

```
$ docker login
$ docker push lleonini/assignment1
```

- To execute my image

```
$ docker run -p 8080:8080 lleonini/assignment1
Listening on port:  8080
```

DO YOUR OWN VERSION

- Using Python
- Google: "Python minimal http server"
- Then bundle it in a Docker image

COMMIT YOUR ASSIGNMENT

```
$ cd ..
$ git add assignment1
$ git commit -m "assignment1 finished"
$ git push
```