Fiji NFMS ER Docker Summary

Eamon Green, Michael Green

Nov 2021

Table of Contents

# Docker Setup Course

This will repo go over the instructions and examples for the Docker Course.

## information

This course goes over “3 days” focusing on buidling the users skill to be able to use docker for the current projects intended purpose.

| Title | Link | Details | Example Files |
| --- | --- | --- | --- |
| How to install docker | [**Install Docker**](Fiji-NFMS-ER-Docker-Setup.Rmd) | This goes through setting yourself up to start working with docker on your system | [**Install Docker**](Fiji-NFMS-ER-Docker-Setup.Rmd) |
| Training 1 | [**Training 1**](Fiji-NFMS-ER-Docker-Tutorial1.Rmd) | This goes through setting yourself up to start working with docker on your system | [**Session 1 Examples**](/Example1/examples) |
| Training 2 | [**Training 2**](Fiji-NFMS-ER-Docker-Tutorial2.Rmd) | This takes you through the next steps after setting up docker | [**Session 2 Examples**](/Example2/examples) |
| Training 3 | [**Training 3**](Fiji-NFMS-ER-Docker-Tutorial3.Rmd) | This takes you through the next steps after setting up docker | [**Session 3 Examples**](/Example3/examples) |

## Training 1

### Pre-Requisites

Note: To run docker on Windows 10 Home, you must:

* be operating as the machine administrator user profile
* Have WSL2 Linux Kernel installed and updated [here](https://docs.microsoft.com/en-us/windows/wsl/install-win10)

### Overview:

* <https://hub.docker.com/_/hello-world>
* This checks docker is correctly installed and operational
* Running use command prompt. Windows key + R, cmd opens the command prompt
* Run in command line –> docker run hello-world

### Step 1: Installing Docker

Installing Docker on a windows machine.

1. Follow the link [here](https://docs.docker.com/docker-for-windows/install/) to go to the docker website and download it.
2. Select Download from Docker hub and click on this button (this is a free download) Docker Download for Windows
3. Once the download has been completed open the installer and follow the prompts to completed the installation of Docker. Once installed you will be required to restart your computer before running the Docker software.
4. After the restart, open Docker Desktop. The Docker Whale Symbol should also appear in your bottom toolbar.

### Step 2: Hosting Liferay – Docker Compose

The FIMS is operated on a software platform called Liferay. The next step is to host the Liferay platform through docker using Docker Compose

**Familiarization with Docker Compose and Hosting of Liferay**

1. Open Docker Desktop and command prompt
2. Follow the instructions given on the following link to carry out an exercise to practice using [Docker Compose](https://docs.docker.com/compose/gettingstarted/#step-3-define-services-in-a-compose-file).

**Download Files from Drop Box**

1. Download the docker-compose.yml file, Backup and Volumes folder onto your local directory with the same directory structure as on the dropbox

**Hosting Liferay**

1. Open Docker Desktop
2. Open Command prompt and navigate to the directory (cd prompt) containing the saved files from step 1 (docker-compose, backup, volumes)
3. In command prompt, start the application by running docker-compose up -d
4. Goto [Local Host](http://localhost:8080/) in a browser to see the application running. At this stage this will show the Liferay platform home screen. Part 3 will explain how to install/restore the Forest Information Management System onto the Liferay platform.

### Step 3: Configuration and Installation of FIMS

[**Configuration and Installation of FIMS**](../FIM/FIM.md)

### Step 4: Adding New Users

[**Adding New Users**](../User.md)

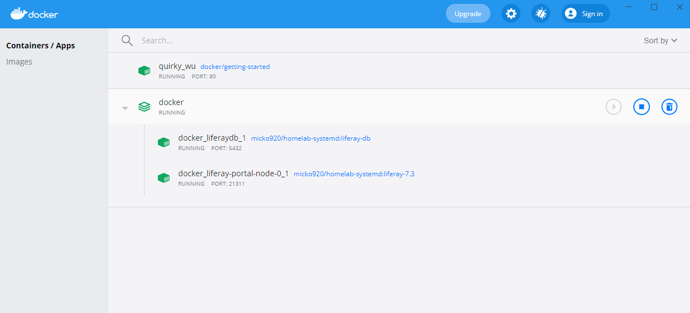
### Step 5: Accessing FIMS

**Access site from another computer on local LAN**

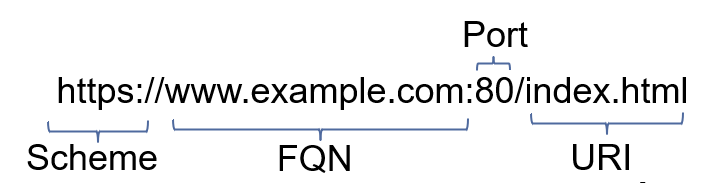
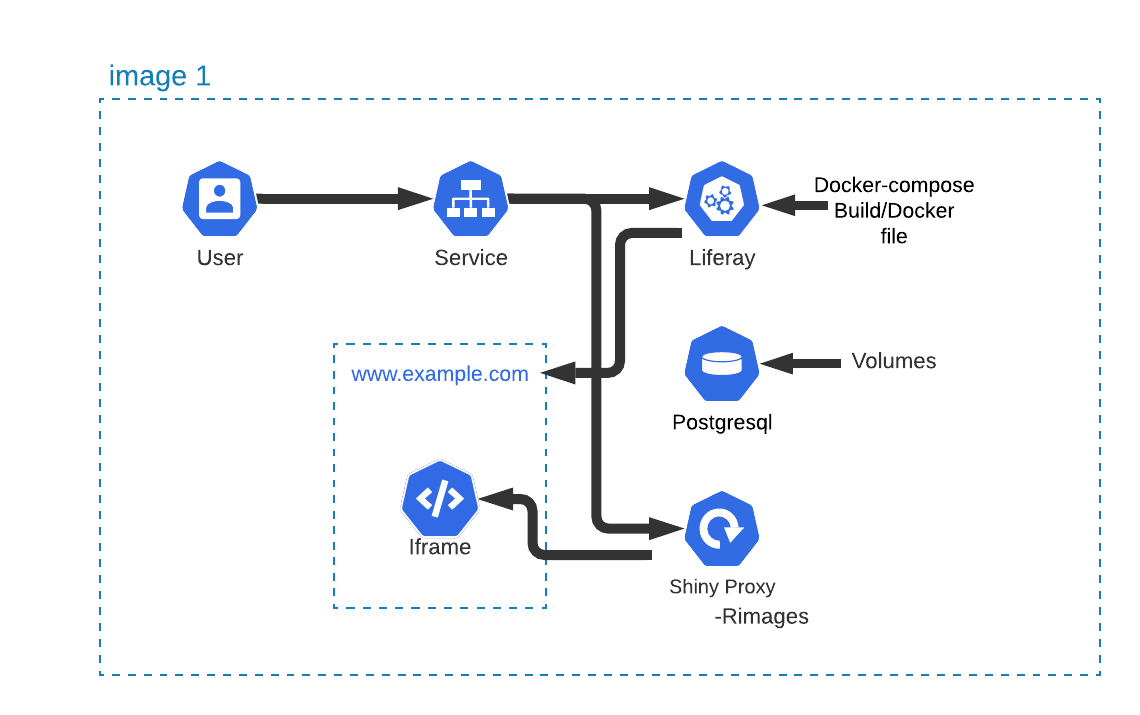
1. Enter IP address of the host computer into browser followed by port 8080. e.g 192.168.x.xx:8080 and confirm FIMS loads. This confirms the site is being hosted on the local LAN.

### Step 6: Ending Docker Session

1. Return to the docker interface and select the ‘Stop’ Button, or in the command prompt Ctrl + C or docker-compose stop command.



### Step 7: FIMS Full Sysyem - End Goal

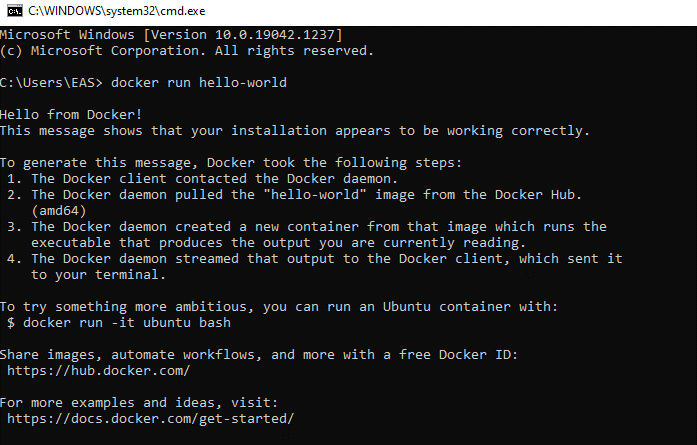
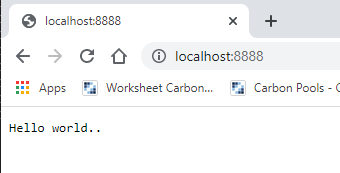
 

## Training 1

## Session info:

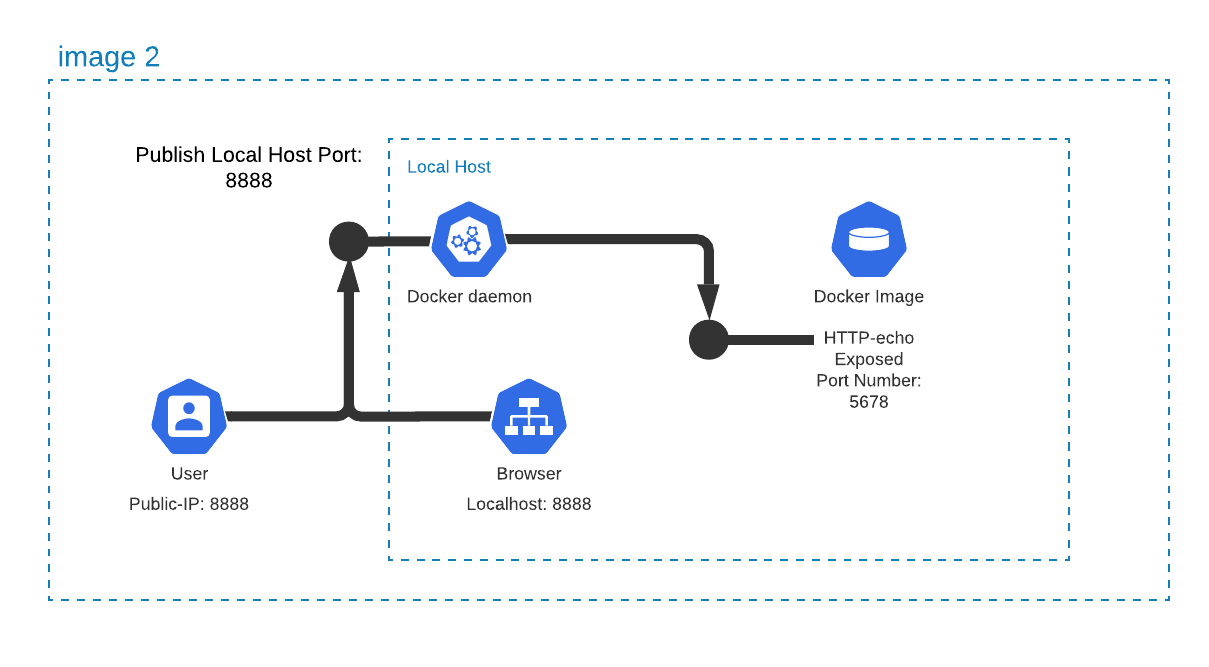
* Download the example files in this directory of the Repo
* Code that inclues ##: are refering to the line number it resides on. Do not copy this into the code.

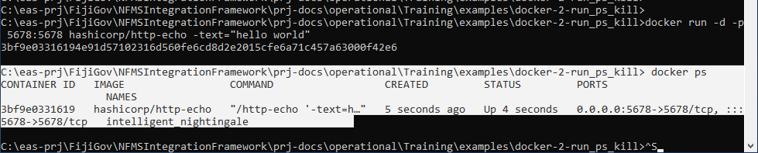
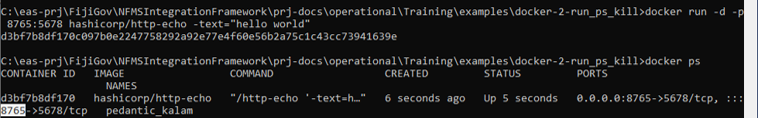
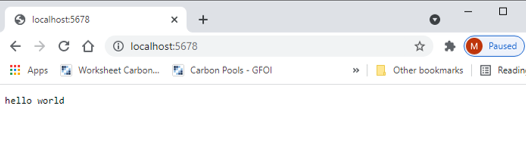
### Docker; 1 - Run

1. Neccesary Commands
   * Pull
   * Image
   * Run
2. Process
   1. Open command prompt
   2. Change directory to directory where the zip file was extracted to (e.g docker-1-run) (cd)
   3. Type -> notepad.exe command.txt to open the command text file
   4. Select first command line with mouse and copy
   5. Paste into command prompt
   6. Where a line starts with a # this is an instruction rather than a command e.g visit a browser and copy in the URL e.g [localhost:5678](https://localhost:5678)
3. Output
   1. Terminal Output 
   2. Web Solution 

### Docker; 2 - Run + PS + Kill

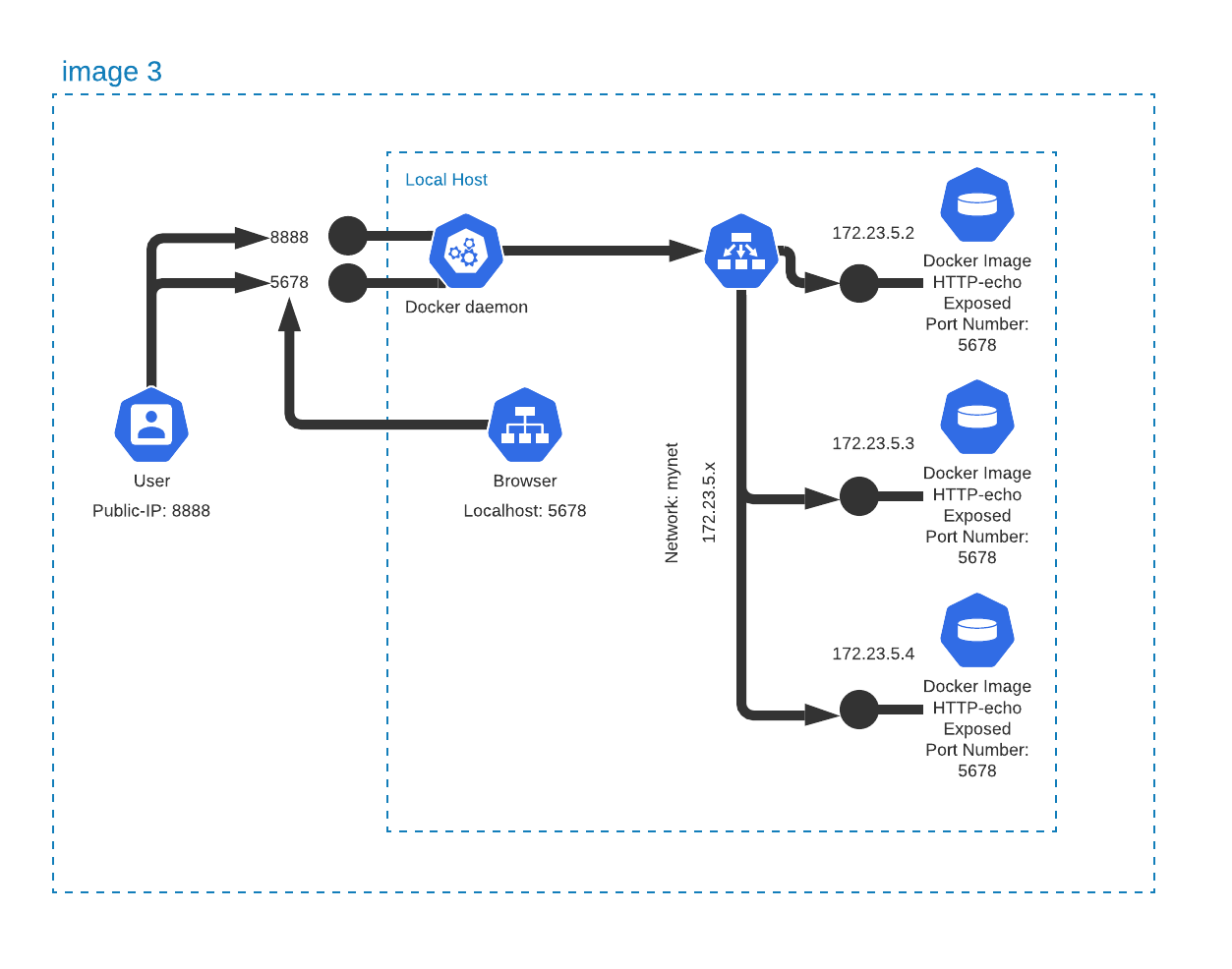
1. Neccesary Commands
   * Run
     + Detach/interactive (-d/-it)
     + Ports - Publish/Expose
   * PS – lists instances that are running on the server
   * Kill – removes running image



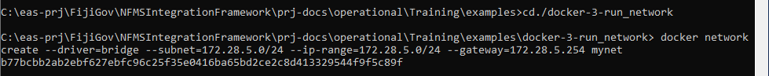
1. Process
   1. docker docker run -d -p 5678:5678 hashicorp/http-echo -text="hello world" 
   2. docker docker run -d -p 8765:5678 hashicorp/http-echo -text="hello world"docker ps 
   3. docker docker run -d -p 8888:5678 hashicorp/http-echo -text="hello world" 

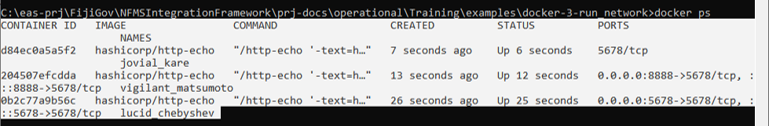
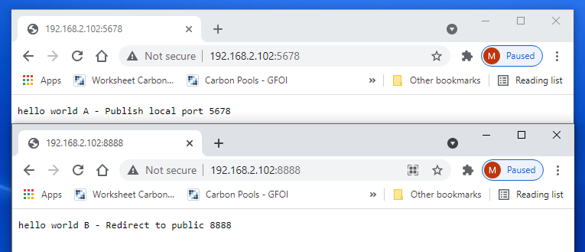
### Docker; 3 - Run + Network

1. Neccesary Commands
   * Run
     + Network



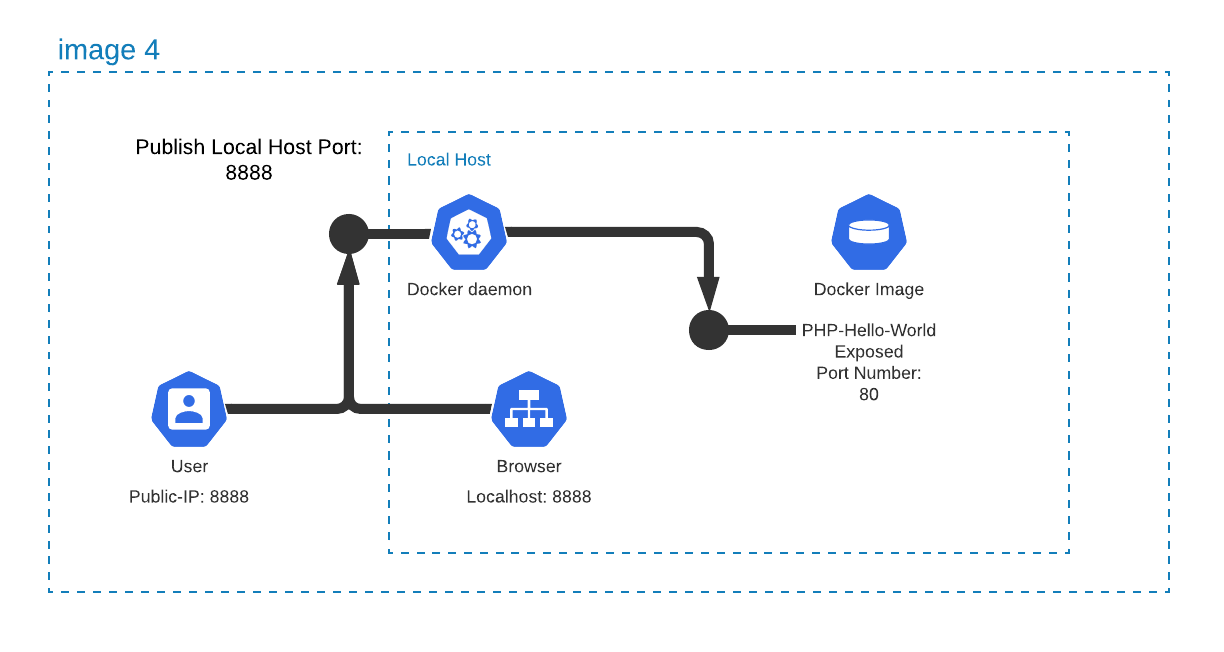
1. Process
   1. docker network create --driver=bridge --subnet=172.28.5.0/24 --ip-range=172.28.5.0/24 --gateway=172.28.5.254 mynet



1. docker run -d -p 5678:5678 --network=mynet --ip="172.28.5.2" hashicorp/http-echo -text="hello world A - Publish local port 5678“
2. docker run -d -p 8888:5678 --network=mynet --ip="172.28.5.3" hashicorp/http-echo -text="hello world B - Redirect to public 8888“
3. docker run -d --expose 5678 --network=mynet --ip="172.28.5.4" hashicorp/http-echo -text="hello world C - Exposed port 5678 only”docker ps 
4. Exposed port accessed through external IP address 

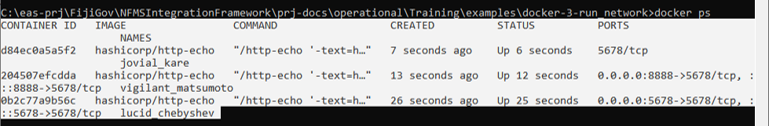
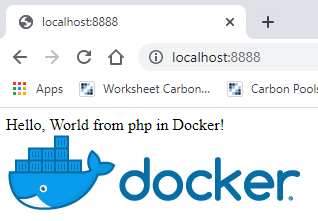
### Docker; 4 - Build

1. Neccesary Commands
   * Build



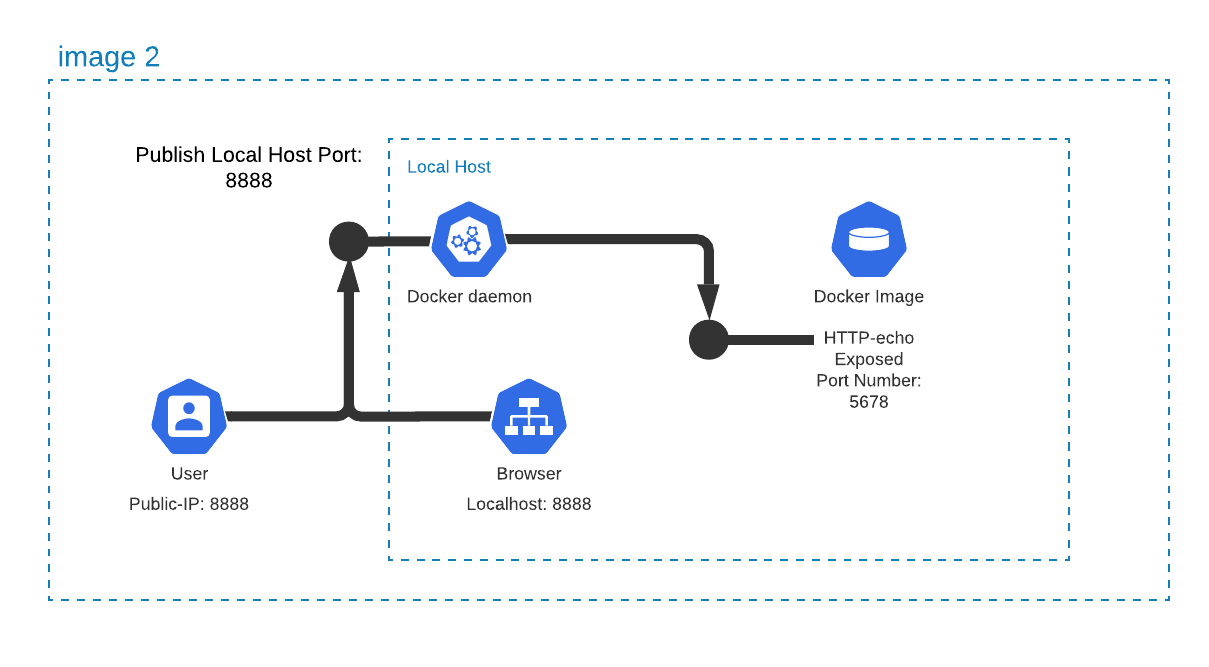
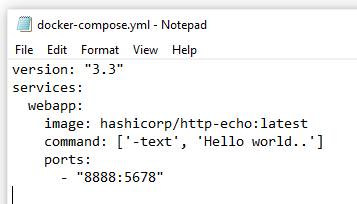
1. Process
   1. docker build -t php-hello-world
   2. docker image
   3. docker run -d -p 8888:80 php-hello-world
   4. docker ps

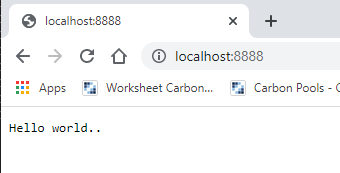
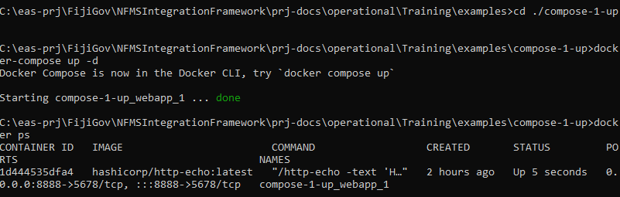
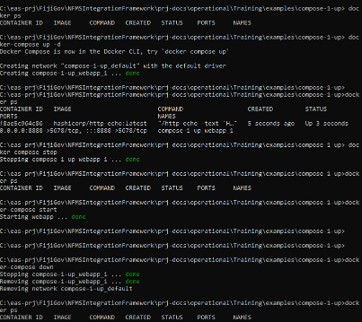
Use your desired browser and enter [localhost:8888](https://localhost:8888) into the search bar

1. docker run -d -p 5678:5678 --network=mynet --ip="172.28.5.2" hashicorp/http-echo -text="hello world A - Publish local port 5678“docker run -d -p 8888:5678 --network=mynet --ip="172.28.5.3" hashicorp/http-echo -text="hello world B - Redirect to public 8888“docker run -d --expose 5678 --network=mynet --ip="172.28.5.4" hashicorp/http-echo -text="hello world C - Exposed port 5678 only”docker ps 
2. Output

### Compose; 1 - Compose + Up

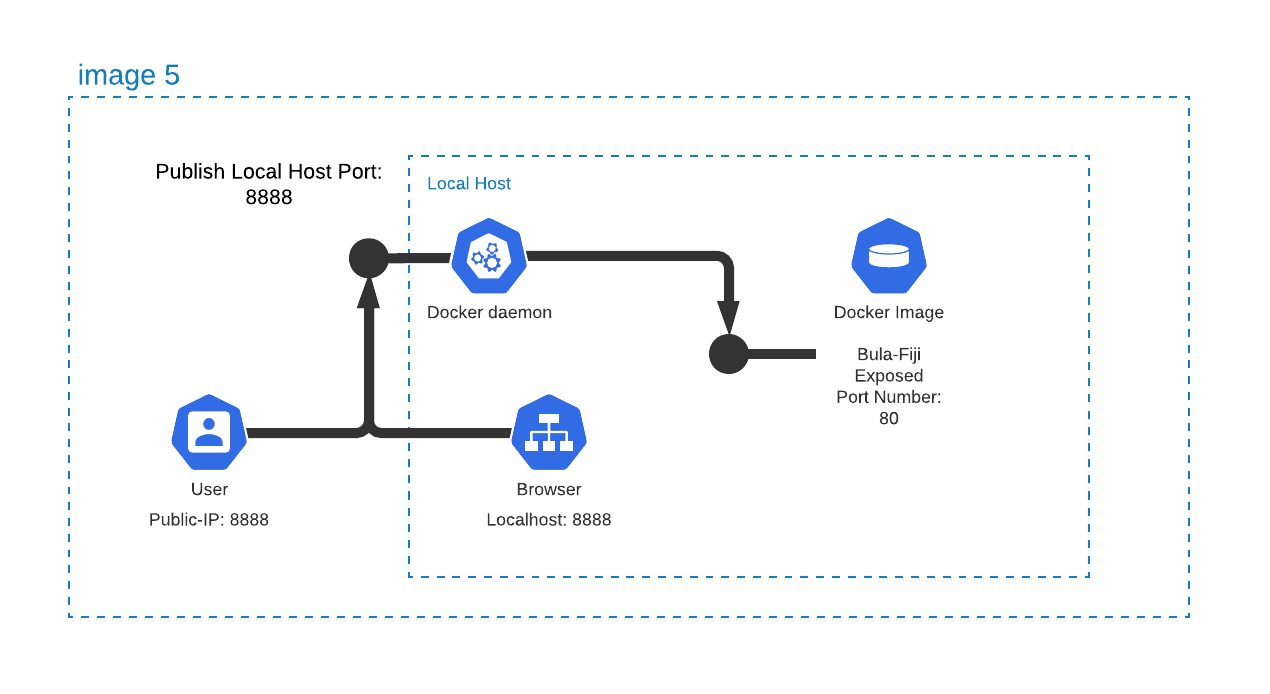
1. Neccesary Commands
   * Up
   * Down
   * Start
   * Stop

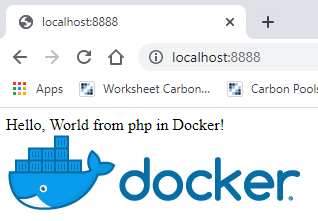
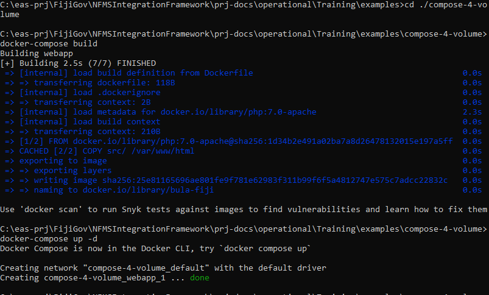
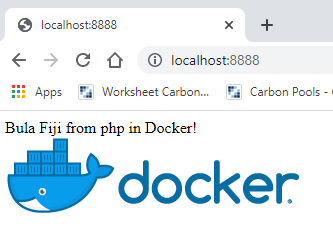
 

1. Process
   1. docker-compose up -d
   2. docker ps
   3. Browse [localhost:8888](https://localhost:8888)
   4. Now up, start, stop, stopdocker-compose up -d
   5. docker ps
   6. docker-compose start
   7. docker ps
   8. docker-compose down
   9. docker ps [//]: # Re do image

### Compose; 2 - Compose + Build

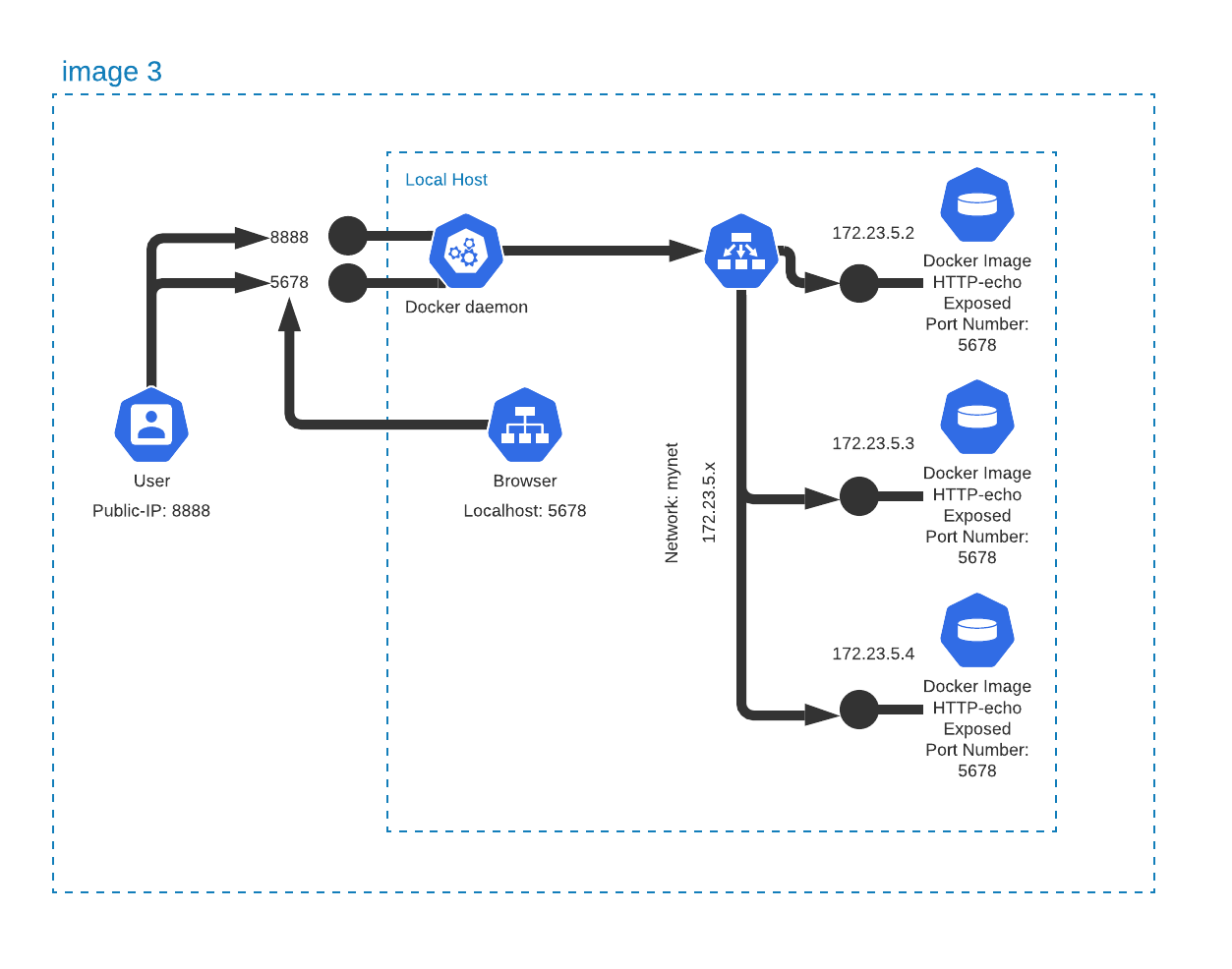
1. Neccesary Commands
   * Up
   * Down
   * Start
   * Stop
   * Build

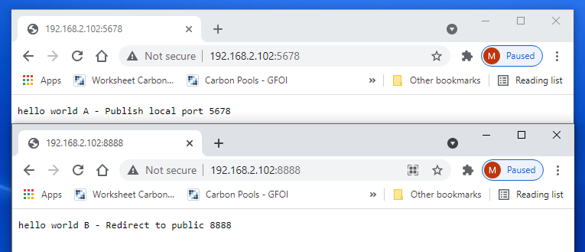
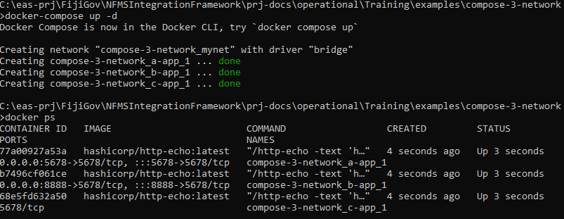
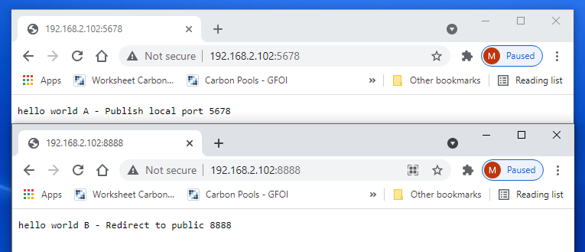


1. Process
   1. docker-compose build
   2. docker-compose up -d
   3. Browse [localhost:8888](https://localhost:8888) Notice there is no “down”
   4. Change the content in bula-fiji/src/index.php to;
   5. 3: echo "Hello, World from php in Docker!<br>";
   6. 3: echo "Bula Fiji from php in Docker!<br>";
   7. Browse [localhost:8888](https://localhost:8888)

### Compose; 3 - Compose + Network

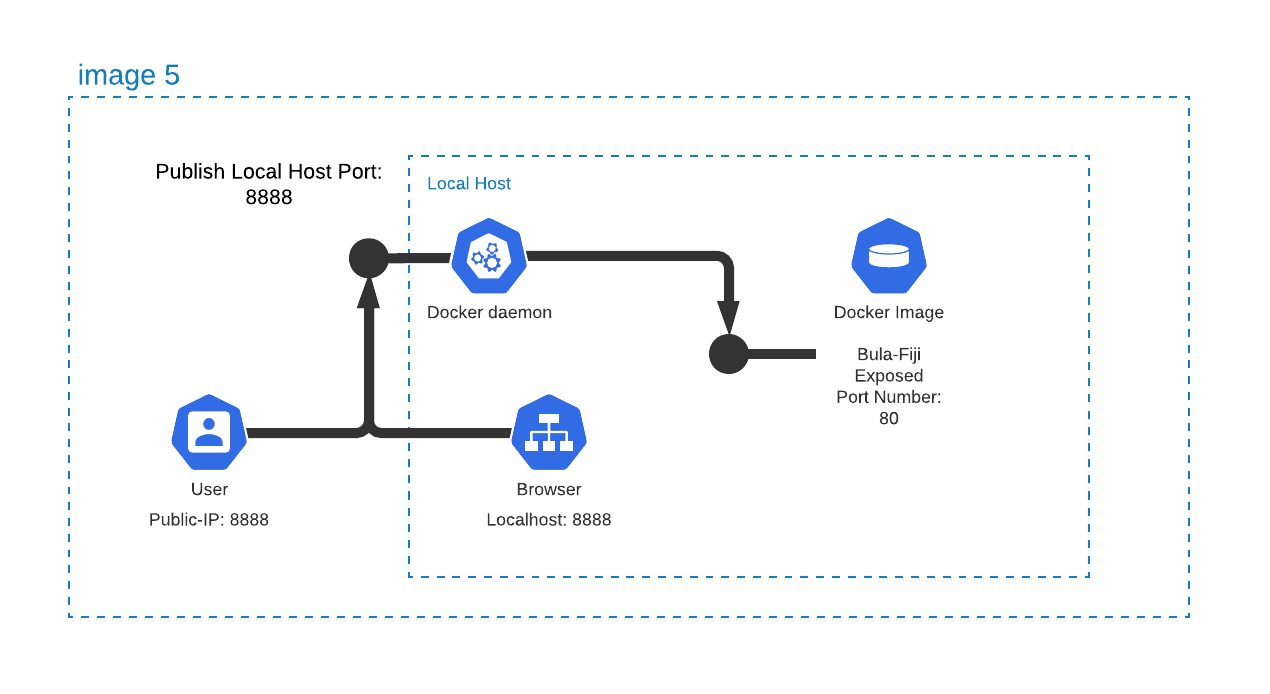
1. Neccesary Commands
   * Up
   * Down
   * Start
   * Stop

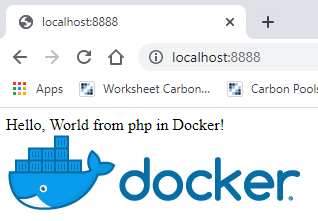
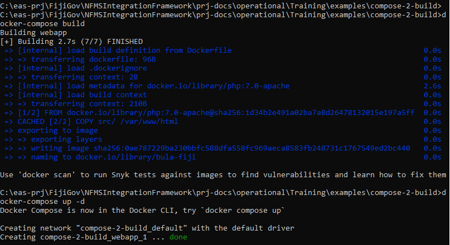
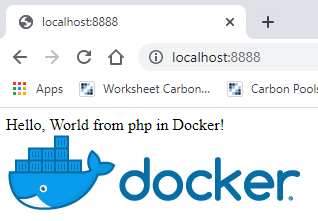


1. Process
   1. docker-compose up -d
   2. docker ps
   3. Browse (A) [localhost:5678](https://localhost:5678)Browse (B) [localhost:8888](https://localhost:8888)
   4. Now open these ports in someone elses browser

### Compose; 4 - Compose + Volume

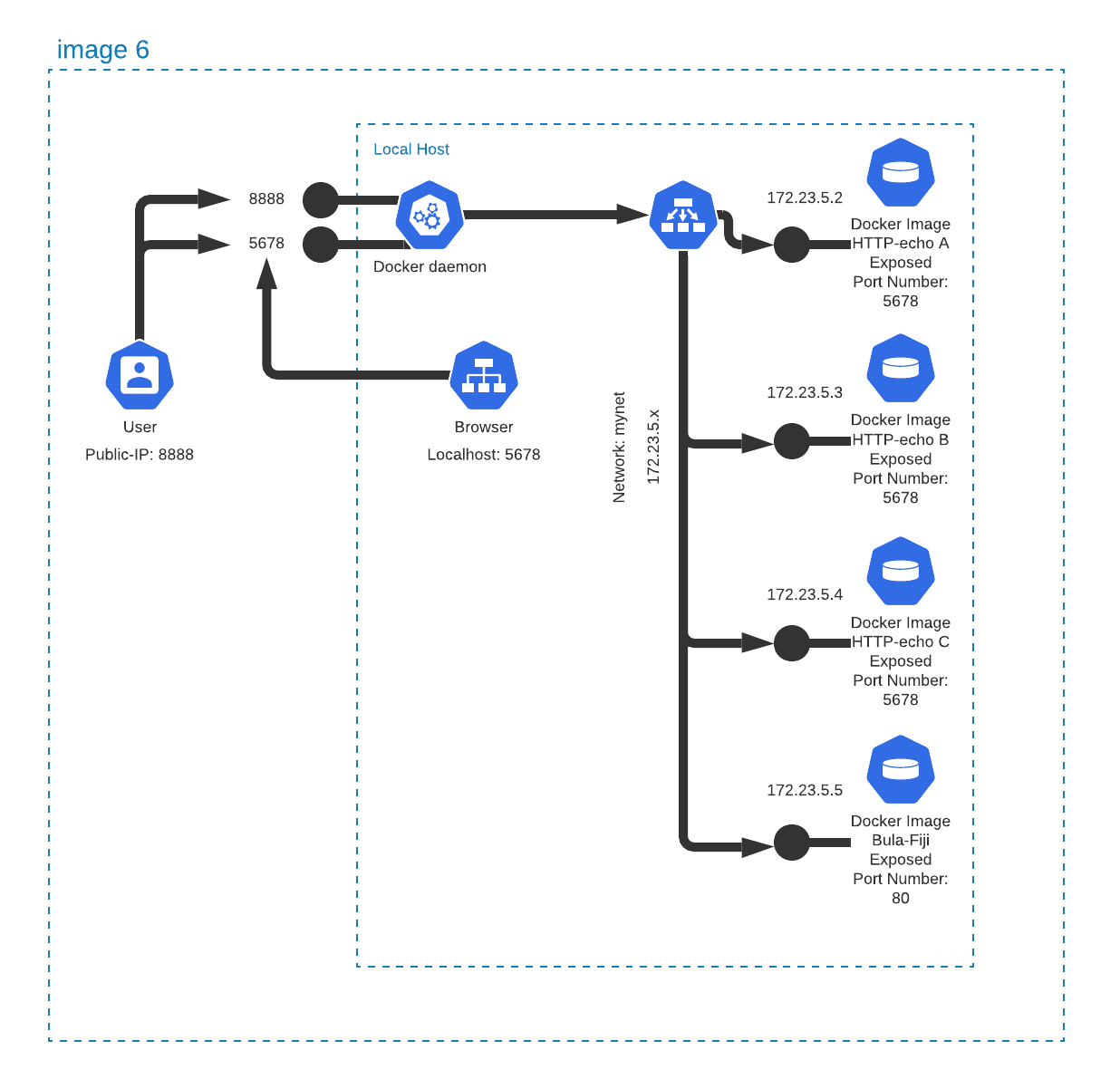
1. Neccesary Commands
   * Up
   * Down
   * Start
   * Stop
   * Build

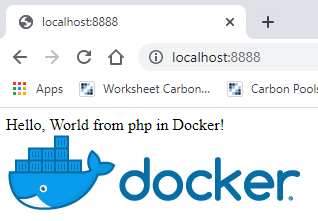
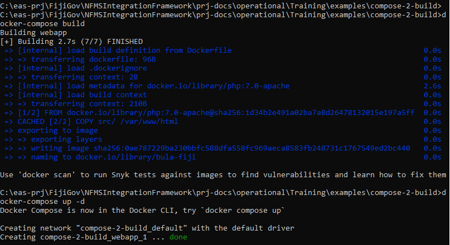
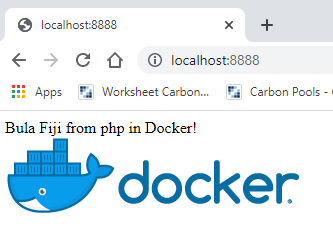


1. Process
   1. docker-compose build
   2. docker-compose up -d
   3. [localhost:8888](https://localhost:8888)
   4. Change the content in volumes/bula-fiji-node/index.php to;
   5. 3: echo "Hello, World from php in Docker!<br>";3: echo "Bula Fiji from php in Docker!<br>";There is no need to rebuild
   6. [localhost:8888](https://localhost:8888)

### Compose; 1 - Skeleton System

1. Neccesary Commands
   * Run
     + Network

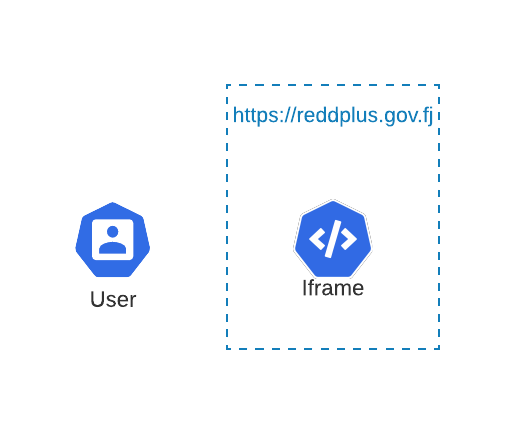


1. Process
   1. docker-compose build
   2. docker-compose up -d
   3. [localhost:8888](https://localhost:8888)
   4. 
   5. Change the content in volumes/bula-fiji-node/index.php to;
   6. 3: echo "Hello, World from php in Docker!<br>";3: echo "Bula Fiji from php in Docker!<br>";There is no need to rebuild
   7. [localhost:8888](https://localhost:8888)

### Terminology

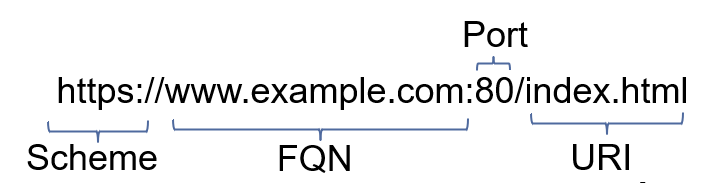
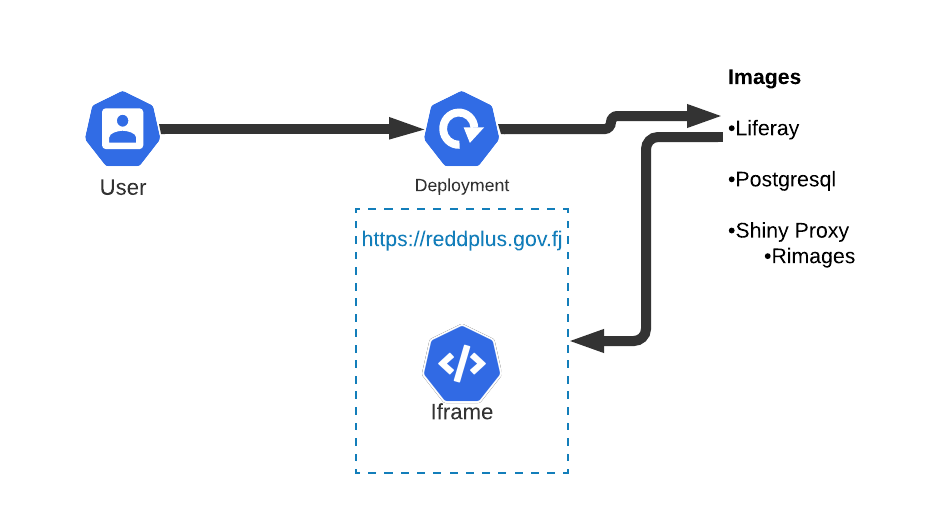
1. Proxy
   * Rev Proxy
   * Switch (Name/IP/Portal)
   * SSL
2. Images
   * Postgresql
   * Liferay
   * Caddy
   * Shiny Proxy
   * Hello World
   * Http Echo

### Skeleton System

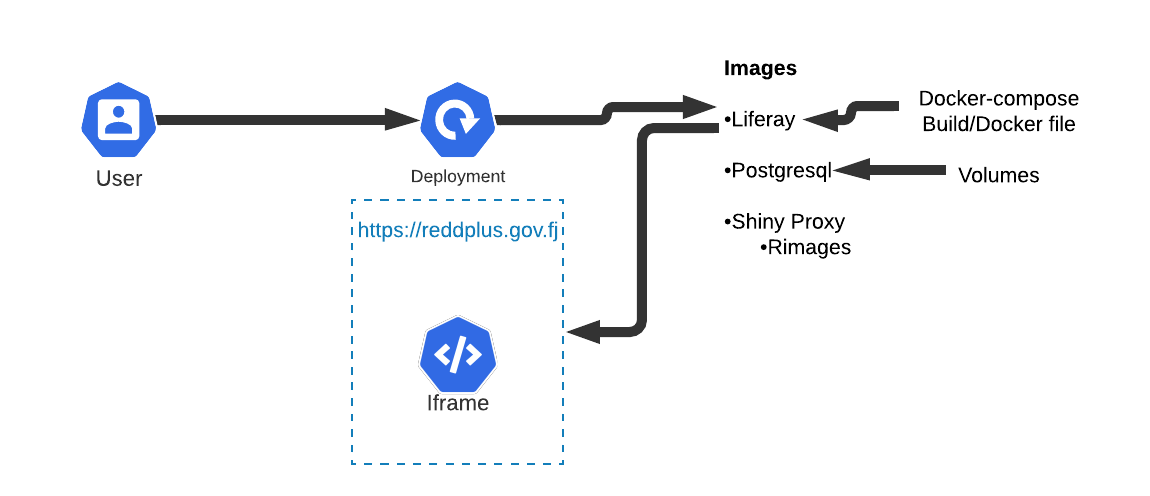


1. Images
   * Postgresql
   * Liferay
   * Shiny Proxy
     + Rimages

### Traffic Routing (inc Liferay)

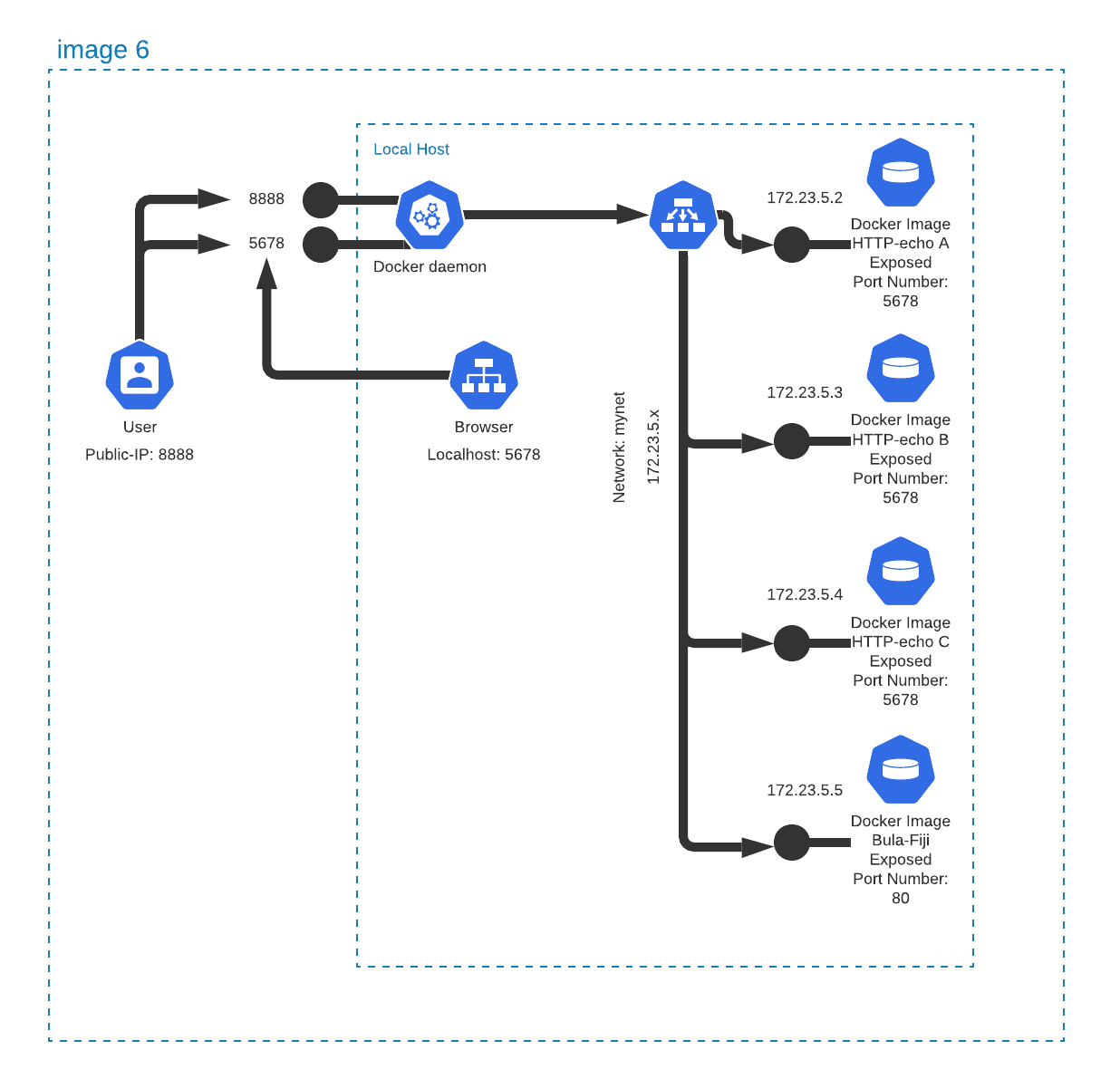
### Traffic Routing (Full Complexity)



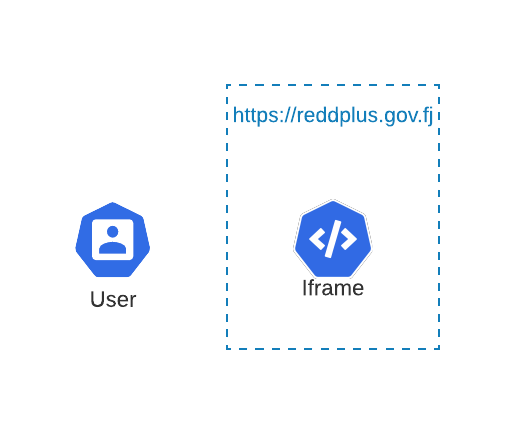
## Training 2

### Skeleton; 1 - System

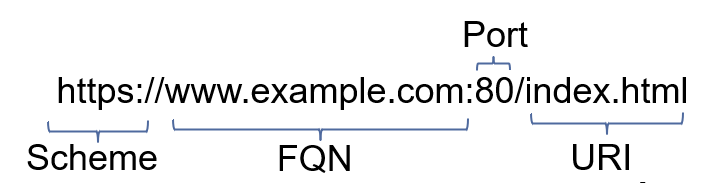
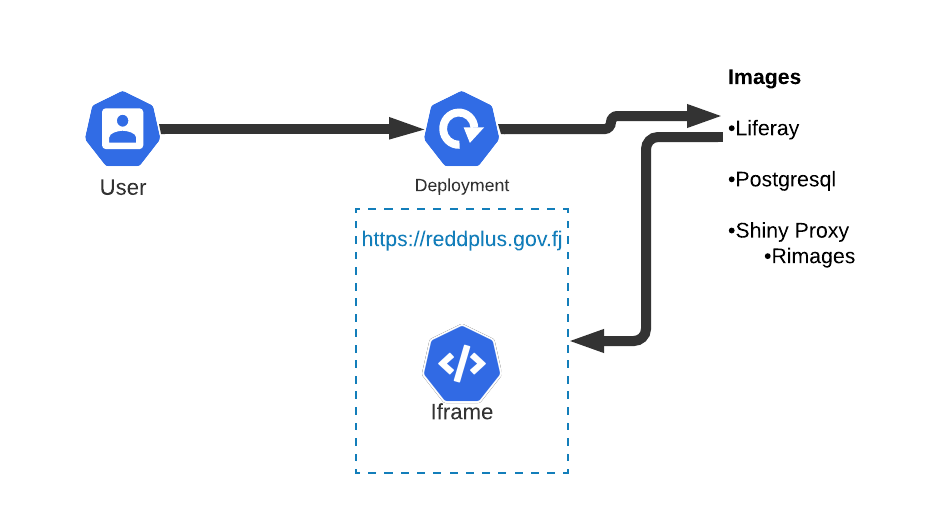
1. Neccesary Commands
   * Run
     + Network



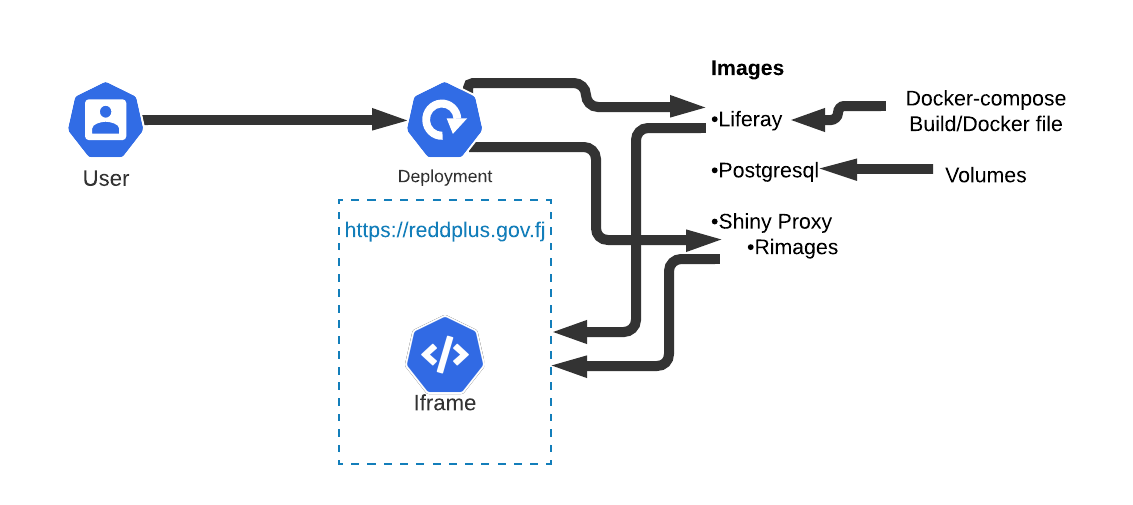
1. Images
   * Lifray
   * Postgresql
   * Shiny Proxy
     + Rimages
2. System



### Traffic Routing (inc Liferay)

### Traffic Routing (Full Complexity)



### Terminology

1. Proxy
   * Rev Proxy
   * Switch (Name/IP/Portal)
   * SSL
2. Images
   * Postgresql
   * Liferay
   * Caddy
   * Shiny Proxy
   * Hello World
   * Http Echo

### Key Learning Outcomes

* Backup
  + Volumes
  + Database
  + Liferay
* Trials and Upgrades
* Skeleton System – (Local System in ITC Context)
  + Firewall
  + Active Directory
  + DNS Names
  + IP/Ports
* Security
  + User Authentication
  + Firewall

## Training 3

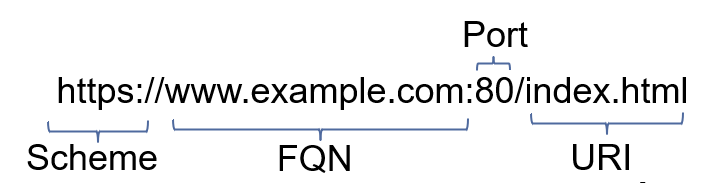
## Objectives

1. Install FIMS
2. Continuation of more advanced Docker
   * Trails
   * Upgrades
   * Backup
3. Production installation plan

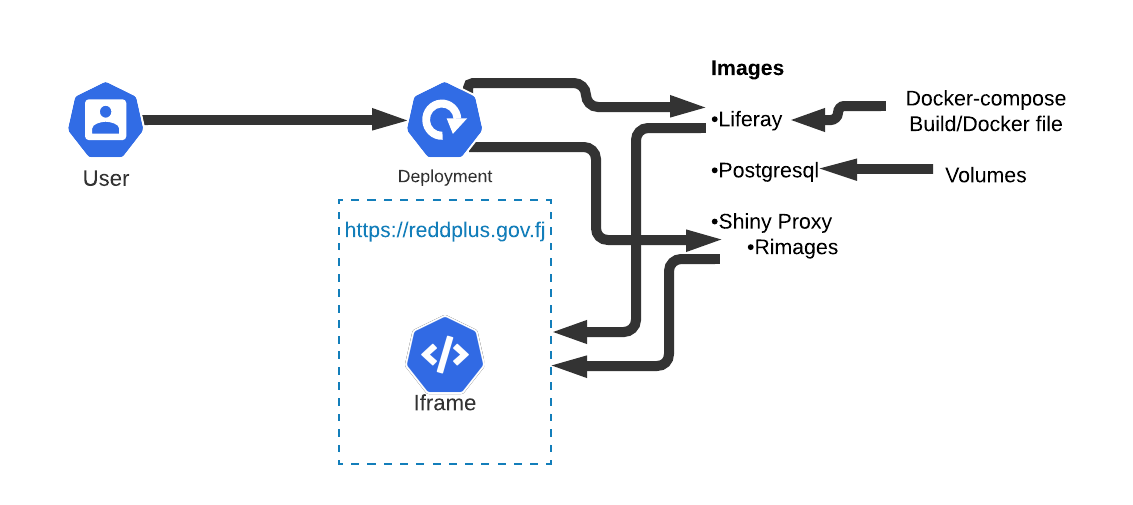
### Install FIMS Liferay

[Install Presentaion](/)

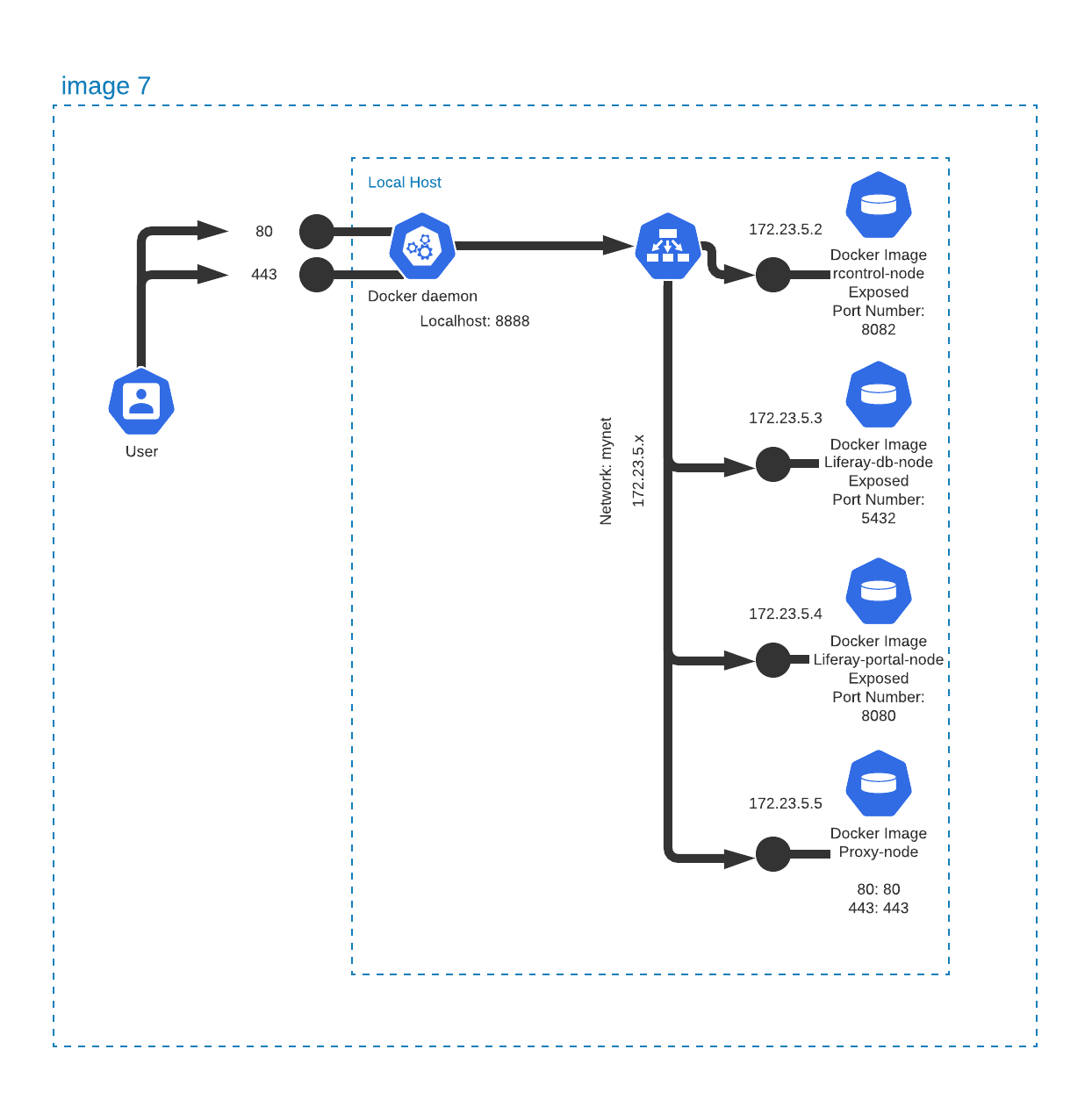
## FIM System Design



### Logical Design

* 

### Physical Connection

* E:-prj-NFMS-ER-Docker-Universe.Rmd
* 

## More Advanced Docker

* Backup: docker-4-volumes
* Upgrades: skeleton-5-upgrades
* Trials (blue/green): skeleton-6-trials

## Docker; 4 - Volumes

### Commands:

1. docker-compose build
2. docker-compose up -d
3. [localhost:8888](https://localhost:8888)
4. docker cp docker-4-volumes\_webapp\_1:/var/www/html .\
5. You should now have a new directory called html, navigate to index.php within the directory and open with your desired editor. (e.g. notepad.exe html/index.php)
6. Change the content from;
   * 3: echo "Hello, World from php in Docker! <br>"; to
   * 3: echo "Bula Fiji from php in Docker! <br>";
7. docker cp .\html docker-4-volumes\_webapp\_1:/var/www/
8. There is no need to rebuild, browse [localhost:8888](https://localhost:8888)
9. docker-compose down

## Skeleton; 5 - Upgrades

### Commands:

1. docker build -t bula-fiji:v1 .\bula-fiji
2. docker-compose up -d
3. docker ps
4. browser (A) [localhost:5678](https://localhost:5678)
5. browser (B) [localhost:8888](https://localhost:8888)
6. browser (bula-fiji) [localhost:8222](https://localhost:8222)
7. docker cp docker-4-volumes\_webapp\_1:/var/www/html .\
8. Using notepad.exe bula-fiji\src\index.php change the content in bula-fiji.php from;
   * 3: echo "Hello, World from php in Docker! <br>"; to
   * 3: echo "Bula Fiji from php in Docker! <br>";
9. docker build -t bula-fiji:v2 .\bula-fiji
10. Change the content in docker-compose.yml from;
    * 28: image: bula-fiji:v1 to
    * 28: image: bula-fiji:v2
11. docker-compose up -d
12. Browse [localhost:8888](https://localhost:8888)
13. docker-compose down

## Skeleton; 6 - Trails

### Commands:

1. docker-compose build
2. docker-compose up -d
3. Browse [localhost:8080](https://localhost:8080)
4. Check that it identifies itself as (A)
5. Now change volumes/proxy-node/caddy/Caddyfile from;
   * 24: reverse\_proxy http://172.28.5.2:5678 { to
   * 24: reverse\_proxy http://172.28.5.2:5678 http://172.28.5.3:5678 {
6. docker-compose restart proxy-node
7. Change the content in docker-compose.yml from;
   * 28: image: bula-fiji:v1 to
   * 28: image: bula-fiji:v2
8. Browse [localhost:8080](https://localhost:8080)Check that it identifies itself as (A) or possibly (B)If you keep refresh the page it will change
9. docker-compose down