DevOps Course – Assignment1 - Report

First of all I want to include this:

You should prepare your system in a way that the course staff can test the system with the following procedure (on Linux). I have my first assignment under DevOps_Course/Assignment1 and on branch “exercise1”. When cloning you get DevOps_Course and the “exercise1” branch should be the selected branch:

```
$ git clone -b exercise1 https://github.com/ju1pp1/DevOps_Course
$ cd DevOps_Course
$ cd Assignment1
$ docker-compose up –-build
... wait for 10s
$ curl localhost:8199/status
$ docker-compose down
```

How to erase data:

Erase storage container data:

```
$ docker compose down
$ docker volume rm $(docker volume ls -q | grep storage_data)
$ docker compose up –d
```

Erase vStorage bind (host):

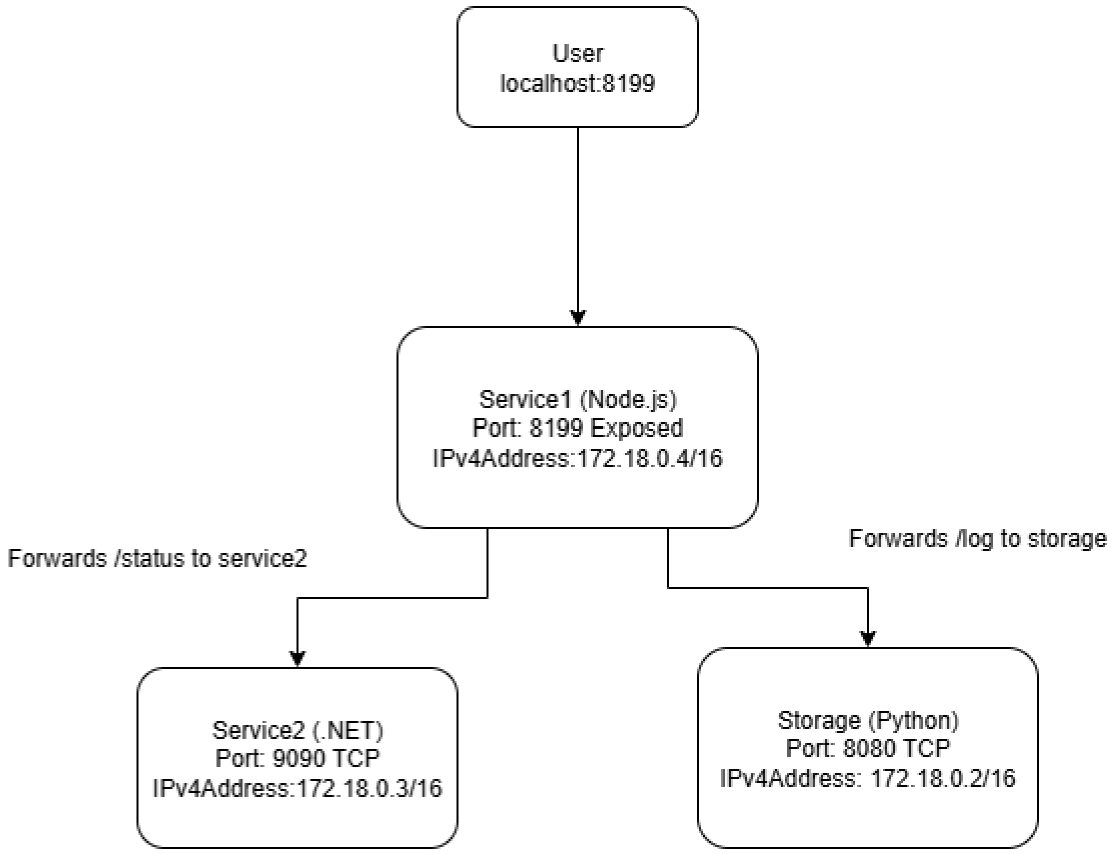
```
$ rm -f ./vstorage/log.txt
```

About the task:

PC, AMD Ryzen 5 7600X3D 6-Core Processor, RAM 32GB

Windows

Docker Desktop 4.44.3, Docker Engine v28.3.2, Docker Compose version v2.39.1-desktop.1



Analysis of the content of the status records. Which disc space and uptime you actually measured? How relevant are those measurements? What should be done better?

Uptime that is measured is container's view of /proc/uptime. It's actually host Kernel's system uptime in Linux, so I think it is not a real service uptime if we e.g. compare it to process uptime. The current one does not reflect how long container or service itself has been running.
Free disk in this implementation is measured as df –Pm / and that means available disk space in container root filesystem. So it reflects to Docker virtual filesystem, not my whole physical disk space.

If we think of relevance, the uptime can be misleading if we want to know when the service has started. If the host machine has been up for 10 days and it was restarted 2 minutes ago, it still reports for the 10 days time. Disk space is not measuring application-specific storage e.g. how much space /vstorage or /data are using.

What I could have done better is that maybe the process uptime of the service could have been measured. By this I mean when the main process has started. But perhaps the intention of this assignment was not to do that.

Analysis and comparison of the persistent storage solutions – what is good and what is bad in each solution?

I think the **vStorage** is easy to test and it helps that it is visible to host even if storage is stopped. In addition to that when restarted the logs survive through that and keep the logs done earlier.
One bad thing I noticed is that it exposes host filesystem that can be a security risk. I think it also breaks encapsulation, if the whole idea of the container is that they should be self-contained and portable.

Storage container was the named volume which is decoupled from the host and it also persists across restarts. It is isolated unlike vStorage, so if we stop Storage and update logs the vStorage logs will update, but not Storage log. It is fully managed by Docker, it abstracts the paths and storage backend. It can be slightly harder to inspect and clean than vStorage.

What was difficult?

For me understanding how to make the service2 and storage to be internal and service1 forwards the GET to service2 status and POST to storage. Making sure only service1 is externally accessible. Also, the calculation of the Mbytes for the status.

What were the main problems?

In my own implementation I had issues with .NET runtime image missing the Microsoft.NET.Sdk.Web, which was the key to get the C# (.NET) implementation running. Alongside that as what I said earlier, to make the service1 forward requests for service2 and storage and make those unable to be opened manually.