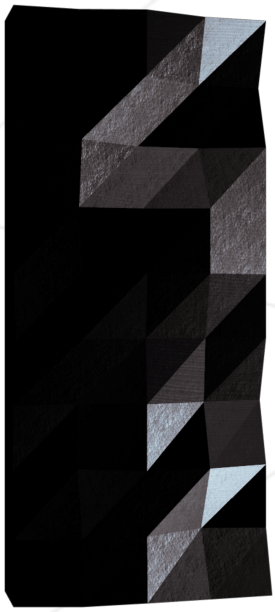

Ft_services

Summary: This is a System Administration and Networking project.

Contents

Chapter I

Introduction

Ft_services will introduce you to kubernetes. This project aims to deepen your knowledge about using Docker with docker-compose and will let you discover cluster management and deployment with Kubernetes. You will virtualize a network and do "clustering".

Chapter II

General instructions

- You must put all the necessary files for the configuration of your server in a folder called `srcs`.
- Your `setup.sh` file should be at the root of your repository. This script will setup all your applications.
- This subject requires both old and new practices. We therefore advise you not to be afraid to read a lot of documentation about docker, kubernetes, and all other things useful for the project.

Chapter III

Mandatory part

The project consists of setting up an infrastructure of different services. To do this, you must use `Kubernetes`. You will need to set up a multi-service `cluster`.
`Each service will have to run in a dedicated container`.
Each container `must` bear the same name as the service concerned and the containers have to be build using Alpine Linux.

You will also have to set up:

- The Kubernetes web dashboard. This will help you manage your cluster.
- The `Ingress Controller` which manages the external access of your services. The `Ingress Controller` will redirect to your Nginx container.
- A `Nginx` server listening on ports 80 and 443.
- A `FTPS` server listening on port 21.
- A `WordPress` website listening on port 5050, which will work with a MySQL database. Both services have to run in separate containers. The WordPress website will have several users and an administrator.
- `Phpmyadmin`, listening on port 5000 and linked with the MySQL database.
- A `grafana` platform, listening on port 3000, linked with an `influxDB` database. Grafana will be monitoring `all` your containers. You must create one dashboard per service. InfluxDB and grafana will be in two distincts containers.
- In case of a crash or stop of one of the two database containers, you will have to make shure the data persist.
- You must be able to access the Nginx container by logging into `SSH`.
- All your containers must restart in case of a crash or stop.