



UNIVERSIDAD DE MÁLAGA



Grado en Ingeniería Informática

Generación automática de ficheros de configuración para el
despliegue de servicios en Docker

Automatic generation of configuration files for deploying
services in Docker.

Realizado por
Marcos Domínguez Moreno

Tutorizado por
José Miguel Horcas Aguilera

Departamento
Lenguajes y Ciencias de la Comunicación

MÁLAGA, Octubre de 2024



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADA/O EN TECNOLOGÍAS DE LA INFORMACIÓN

Generación automática de ficheros de configuración para el despliegue de servicios en Docker

**Automatic generation of configuration files for deploying
services in Docker**

Realizado por
Marcos Domínguez Moreno

Tutorizado por
José Miguel Horcas Aguilera

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, OCTUBRE DE 2024

Fecha defensa: Diciembre de 2024

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Keywords: A, B, C

Resumen

En la actualidad, el desarrollo de software y la implementación de servicios en la nube han experimentado un crecimiento exponencial. Las empresas buscan constantemente mejorar sus procesos de desarrollo y despliegue para ofrecer productos y servicios de alta calidad en el menor tiempo posible. En este contexto, herramientas como Docker y metodologías como la Integración Continua (CI) se han convertido en componentes esenciales para lograr estos objetivos.

Docker permite la creación de contenedores que encapsulan aplicaciones y sus dependencias, asegurando que el software se ejecute de manera consistente en cualquier entorno. Por otro lado, la Integración Continua facilita la detección temprana de errores mediante la automatización de pruebas y despliegues, lo que contribuye a un ciclo de desarrollo más ágil y eficiente.

Este trabajo de fin de grado se centra en la generación automática de archivos de configuración para el despliegue de servicios en Docker. El objetivo principal es desarrollar una herramienta que simplifique y automatice el proceso de configuración, permitiendo a los desarrolladores centrarse en la creación de funcionalidades y mejoras en lugar de en tareas repetitivas y propensas a errores.

A lo largo de este documento, se presentará una revisión de la literatura relacionada con Docker, Integración Continua y generación automática de configuraciones. Además, se describirá el diseño y la implementación de la herramienta propuesta, así como los resultados obtenidos y las conclusiones derivadas del proyecto.

Palabras clave: SPL, Docker , Despliegues Servicios , Desarrollo web, Integración Continua , Configuración , feature models , Linea de Productos

Índice

1. Introduction	9
1.1. Motivación	9
1.2. Objetivos	10
1.3. Resultados Esperados	10
1.4. Estructura del documento	10
2. Estado del Arte	11
2.1. Contenedores	11
2.2. imagenes	11
2.3. Docker	11
2.3.1. Dockerfile	12
2.3.2. Buenas Prácticas	12
2.3.3. Problematica	12
2.4. Alternativas Actualmente Disponibles	12
2.4.1. Docker init	12
2.4.2. Docker Extension for Visual Studio Code	13
2.4.3. Inteligencia Artificial	13
2.4.4. phpdocker	13
2.4.5. Starter	13
2.5. Conclusiones	13
2.6. Que se aporta en este contexto:	14
3. Tecnologías Empleadas	16
3.1. UVL	16
3.1.1. Feature Models	16
3.2. Docker	16
3.2.1. Docker Desktop	16
3.3. Node	16

3.4.	Angular	16
3.4.1.	Angular Material	16
3.4.2.	Monaco Editor	16
3.4.3.	ngx-json-viewer	16
3.5.	Nginx	16
3.5.1.	Reverse Proxy	16
3.5.2.	Web Server	16
3.6.	Python	16
3.6.1.	Flask	16
3.6.2.	Pypi	16
3.7.	Git	16
3.8.	GitHub	16
3.8.1.	GitHub Public Api	16
3.9.	Visual Studio Code	17
3.9.1.	UVLS extension	17
4.	Metodología del trabajo. (Fases) Implementación:	19
4.1.	Enfoque/Procedimiento/Justificación/Limitaciones	19
4.2.	Estudio del Estado del Arte :	19
4.3.	Investigación opciones de configuración para Docker	19
4.4.	Requisitos de la aplicación	19
4.5.	Desarrollo Modelos plantillas dockerfile y dockerignore	19
4.6.	Desarrollo del la aplicación de uvengine	19
4.7.	Desarrollo de la web	19
4.8.	Desarrollo de los servicios de backend	19
4.9.	Elaboración de la guía de uso	19
4.10.	Elaboración de un manual de instalación	19
4.11.	Elaboración de la memoria	19
5.	Arquitectura y Modelado del sistema	21
5.1.	Vision general de la arquitectura	21
5.2.	Arquitectura de backend	21

5.3. Arquitectura de Frontend	21
6. Despliegue de la aplicación	23
6.1. Despliegue local	23
6.2. Despliegue en la nube	23
7. Resultados	25
7.1. Resultados Obtenidos	25
7.2. Viabilidad del proyecto	25
8. Conclusiones y líneas Futuras	27
8.1. Conclusion Personal,	27
8.2. Conclusiones del Desarrollo del proyecto	27
8.3. Contribuciones / Aplicaciones Prácticas	27
8.4. Desarrollo de nuevas plantillas	27
8.5. Continuación en el desarrollo de plantillas ya existentes	27
8.6. Mejorar la experiencia del usuario	27
Apéndice A. Manual de Instalación	29
Apéndice B. Manual de Uso	31

1

Introduction

1.1. Motivación

Las herramientas de automatización de tareas son fundamentales en el desarrollo de software, ya que permiten mejorar la eficiencia, la calidad y la consistencia en la producción de código. Al relegar tareas repetitivas y propensas a errores a herramientas automatizadas, los desarrolladores pueden centrarse en tareas más creativas y de mayor valor añadido. Parte del ciclo de vida en un desarrollo de software implica la configuración y el despliegue de servicios en entornos de producción y desarrollo. La configuración y despliegue requiere de la creación de archivos de configuración específicos, como los archivos Dockerfile y .dockerignore, que definen cómo se construye y se ejecuta una aplicación en un contenedor de Docker. En el contexto de la tecnología de contenedores, Docker se ha convertido en una de las plataformas más populares para la creación, el despliegue y la gestión de aplicaciones en entornos containerizados. Sin embargo, la configuración de servicios en Docker puede ser una tarea compleja y propensa a errores, especialmente cuando se manejan múltiples servicios y configuraciones personalizadas. La motivación para realizar este proyecto surge de la necesidad de simplificar y automatizar el proceso de generación de archivos de configuración para el despliegue de servicios en Docker. En la actualidad, la creación manual de estos archivos dockerfile puede ser una tarea tediosa y propensa a errores al no conocer el usuario de toda la información necesaria para la correcta configuración de los servicios. Al desarrollar una herramienta que automatice este proceso, se busca mejorar la eficiencia y la precisión en la configuración de entornos de contenedores, facilitando así el trabajo de desarrollo y administración.

1.2. Objetivos

El objetivo principal de este proyecto es desarrollar una herramienta que permita la generación automática de archivos de configuración para el despliegue de servicios en Docker.

1.3. Resultados Esperados

Dada la gran cantidad de opciones de configuración disponibles en para la generacion de ficheros de configuracion como puedes ser dockerfile..., el desarrollo de esta herramienta presenta varios desafíos técnicos y conceptuales. El primero de ellos es la magnitud de las opciones de configuración disponibles, que pueden variar en función de las necesidades y los requisitos de cada servicio. dar una covertura a todos los escenarios posibles es una tarea inabarcable, por lo que se ha optado por centrarse en los casos de uso más comunes y en las configuraciones más utilizadas en la práctica. Sin embargo se espera que la herramienta sea lo suficientemente flexible y extensible como para permitir la incorporación de nuevas plantillas y configuraciones en el futuro.

1.4. Estructura del documento

Por definir

Estado del Arte

Para la realización de este proyecto se ha realizado un estudio del estado del arte en el ámbito de la generación de archivos de configuración para el despliegue de servicios en Docker. En este apartado se presentan las principales herramientas y tecnologías disponibles en la actualidad, así como las tendencias y los enfoques más comunes en la configuración de servicios en entornos de contenedores.

2.1. Contenedores

Los contenedores son paquetes ligeros que incluyen el código de las aplicaciones junto con sus dependencias, como versiones concretas de entornos de ejecución de ciertos lenguajes de programación y bibliotecas indispensables para ejecutar los servicios de software.

2.2. imágenes

Las imágenes de contenedores son plantillas de solo lectura que contienen el código de la aplicación, las bibliotecas, las dependencias y otros archivos necesarios para ejecutar un contenedor.

2.3. Docker

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Docker es una plataforma de código abierto que permite a los desarrolladores crear, desplegar y ejecutar aplicaciones en contenedores.

2.3.1. Dockerfile

El archivo Dockerfile es un archivo de texto que contiene una serie de instrucciones que Docker utilizará para construir una imagen de contenedor. Usualmente la construcción de un archivo de configuración se realiza a partir de una imagen base, que puede ser una imagen oficial de Docker o una imagen personalizada creada por el usuario. A dicha imagen se añaden las instrucciones necesarias para instalar las dependencias y configurar el entorno de ejecución de la aplicación, lo que viene a representar el contenido de un archivo Dockerfile.

2.3.2. Buenas Prácticas

Para la creación de los ficheros dockerfile es recomendable seguir una serie de buenas prácticas para garantizar la eficiencia y la seguridad en el despliegue de servicios en Docker. Estas prácticas se recogen en la documentación oficial de Docker y en guías de buenas prácticas de la comunidad de desarrolladores.

2.3.3. Problemática

La creación de los ficheros dockerfile puede ser una tarea tediosa y propensa a errores, especialmente cuando se manejan múltiples servicios y configuraciones personalizadas.

2.4. Alternativas Actualmente Disponibles

En la actualidad existen varias alternativas para la generación de archivos de configuración para el despliegue de servicios en Docker (dockerfile)

2.4.1. Docker init

Docker init es el comando de Docker Desktop CITA que permite inicializar un proyecto con un archivo de configuración básico. su funcionamiento consta de elegir una plantilla de proyecto y seleccionar las opciones de configuración deseadas. Las opciones de configuración se limitan a elegir el lenguaje de programación y que comando se ejecutara al iniciar el contenedor.

2.4.2. Docker Extension for Visual Studio Code

La extension de Docker para visual estudio code CITA permite entre sus funcionalidades la generacion de los ficheros dockerfile y dockerignore. su comportamineto es similar al de Docker init, permitiendo elegir el lenguaje de desarrollo del proyecto y que comando se ejecutara al iniciar el contenedor.

2.4.3. Inteligencia Artificial

En la actualidad existen soluciones que se apoyan en la inteligencia Artificial para la generacion de archivos de configuracion, y para la creacion de los ficheros dockerfile y dockerignore. La calidad de estas soluciones es variable y depende de la calidad de los modelos de lenguaje natural empleados y de la calidad del prompt de entrada. como <https://magickpen.com/templates/128/> CITA o alternativas con proposito más general como chatgpt CITA

2.4.4. phpdocker

Tambien estan disponibles alternativas más especificas como <https://phpdocker.io/> CITA que permiten la generacion de archivos de configuracion para servicios en php.

2.4.5. Starter

herramienta open source como estarter CITA <https://www.startwithdocker.com/> capaces de reconocer el lenguaje de programacion del proyecto y generar un archivo de configuracion basico.

2.5. Conclusiones

En este estudio se han recogido algunas de las principales herramientas y tecnologías disponibles en la actualidad para la generación de archivos de configuración. Todas las herramientas proponen mediante plantillas la generacion de los ficheros dockerfile y dockerignore, estas plantillas son limitadas y no permiten la personalizacion de las opciones de configuracion. No obstante suponen un punto de partida para el desarrollo de aplicaiones

2.6. Que se aporta en este contexto:

Las opciones actualmente disponibles para la generacion de archivos de configuracion para el despliegue de servicios en Docker son limitadas y no permiten la personalizacion de las opciones de configuracion. Se deja de lado la variabilidad y las posibles opciones que existen a la hora de construir las imagenes

3

Tecnologías Empleadas

3.1. UVL

3.1.1. Feature Models

3.2. Docker

3.2.1. Docker Desktop

3.3. Node

3.4. Angular

3.4.1. Angular Material

3.4.2. Monaco Editor

3.4.3. ngx-json-viewer

3.5. Nginx

3.5.1. Reverse Proxy

3.5.2. Web Server

3.6. Python

3.6.1. Flask

3.6.2. Pypi

3.7. Git

16

3.8. GitHub

3.8.1. GitHub Public Api

felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

3.9. Visual Studio Code

3.9.1. UVLS extension

EXTENSIONES VISUAL ESTUDIO -UVLS <https://github.com/Universal-Variability-Language/uvl-lsp> -Docker -Flama <https://marketplace.visualstudio.com/items?itemName=diversolab.flamapy> -Graphviz Inetractive Preview <https://marketplace.visualstudio.com/items?itemName=tintinweb.graphviz-interactive-preview> -Latex Worshop -Pylance -Python -Python Debugger -Better Jinja

4

Metodologia del trabajo. (Fases) Implementación:

- 4.1. Enfoque/Procedimiento/Justificación/Limitaciones**
- 4.2. Estudio del Estado del Arte :**
- 4.3. Investigación opciones de configuración para Docker**
- 4.4. Requisitos de la aplicación**
- 4.5. Desarrollo Modelos plantillas dockerfile y dockerignore ...**
- 4.6. Desarrollo del la aplicación de uvengine**
- 4.7. Desarrollo de la web**
- 4.8. Desarrollo de los servicios de backend**
- 4.9. Elaboración de la guía de uso**
- 4.10. Elaboración de un manual de instalación**
- 4.11. Elaboración de la memoria**

5

Arquitectura y Modelado del sistema

- 5.1. Vision general de la arquitectura
- 5.2. Arquitectura de backend
- 5.3. Arquitectura de Frontend

6

Despliegue de la aplicación

- 6.1. Despliegue local
- 6.2. Despliegue en la nube

7

Resultados

7.1. Resultados Obtenidos

7.2. Viabilidad del proyecto

8

Conclusiones y líneas Futuras

- 8.1. Conclusion Personal,
- 8.2. Conclusiones del Desarrollo del proyecto
- 8.3. Contribuciones / Aplicaciones Prácticas
- 8.4. Desarrollo de nuevas plantillas
- 8.5. Continuación en el desarrollo de plantillas ya existentes
- 8.6. Mejorar la experiencia del usuario

Apéndice A

Manual de Instalación

Apéndice B

Manual de Uso



UNIVERSIDAD
DE MÁLAGA

| **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga