Informe técnico

ID: ITIS-2024-CAOSD001

Análisis de variabilidad y generación de ficheros de configuración sobre el modelo de características de Kubernetes (extendido)

Enero 2025

Informe Técnico generado como parte del Proyecto **IRIS: Multi-stage configuration of virtualised services for sustainable adaptation of mobile networks (PID2021-122812OB-I00)**. Proyecto de Generación del Conocimiento financiado por Agencia Estatal de Investigación.

Ministerio de Ciencia, Innovación y Universidades
Investigadoras principales:
Lidia Fuentes Fernández y Mercedes Amor Pinilla

Realizado por Enrique López Encinas (personal contratado) y extendido por Brian Alexander Flores López (personal contratado).

Resumen

Este informe técnico presenta un análisis de la variabilidad de los modelos de características en Kubernetes. Aborda la creación de un modelo de características para Kubernetes tanto manual como de forma automatizada, que captura los diferentes componentes y configuraciones posibles dentro de un clúster de Kubernetes, permitiendo la exploración y gestión de diversas combinaciones de características. Este proceso incluye la identificación de características clave, la organización de estas en una estructura jerárquica y la definición de reglas de variabilidad que determinan cómo se pueden combinar las características. Posteriormente, se analiza la variabilidad y las características de Kubernetes, evaluando las configuraciones posibles. El análisis se realiza tanto para el modelo extraído de forma manual como para el resultante del proceso de extracción automatizado. Este análisis incluye el cálculo del número total de configuraciones posibles, la distribución de probabilidad de que se incluya, etc., considerando las combinaciones válidas de características. El análisis de la variabilidad en Kubernetes permite a los administradores de sistemas y desarrolladores tomar decisiones informadas sobre la configuración de sus clústeres, optimizando el uso de recursos y mejorando la eficiencia operativa, permitiendo de esta forma que las configuraciones de Kubernetes puedan adaptarse a una amplia gama de necesidades y entornos, desde implementaciones pequeñas hasta grandes infraestructuras distribuidas. Con este análisis, se facilita la toma de decisiones estratégicas en la configuración y gestión de clústeres Kubernetes, maximizando su rendimiento y adaptabilidad. Finalmente, se incluyen los comandos necesarios para ejecutar los scripts desarrollados en el proyecto. Estos comandos permiten reproducir los experimentos y análisis descritos en el informe, facilitando la verificación y aplicación de los resultados obtenidos y se discuten las implicaciones prácticas de los resultados obtenidos y se sugieren posibles direcciones para futuros trabajos en este campo.

Índice

1. Introducción

Para comprender el alcance del análisis que se realiza en este informe, es necesario conocer de antemano ciertos conceptos y tecnologías, que se proceden a explicar a continuación:

1.1 Kubernetes

Kubernetes es una plataforma de código abierto diseñada para automatizar la implementación, escalado y gestión de aplicaciones en contenedores. Fue desarrollado originalmente por Google y luego donado a la *Cloud Native Computing Foundation* (CNCF).

Kubernetes facilita la gestión de aplicaciones distribuidas y microservicios en entornos de contenedores, proporcionando herramientas para la automatización de despliegues, escalado dinámico, actualizaciones sin interrupciones y alta disponibilidad.

Estos despliegues, y en general, cualquier objeto de Kubernetes, se realizan usando ficheros de configuración llamados manifiestos, escritos en lenguaje YAML. Estos archivos contienen toda la información para crear un objeto, haciendo uso de la API de Kubernetes y siendo validados por esta durante su creación.

En la figura 1 se muestran los elementos que componen Kubernetes y las relaciones entre ellos. A continuación, se explican los más relevantes:

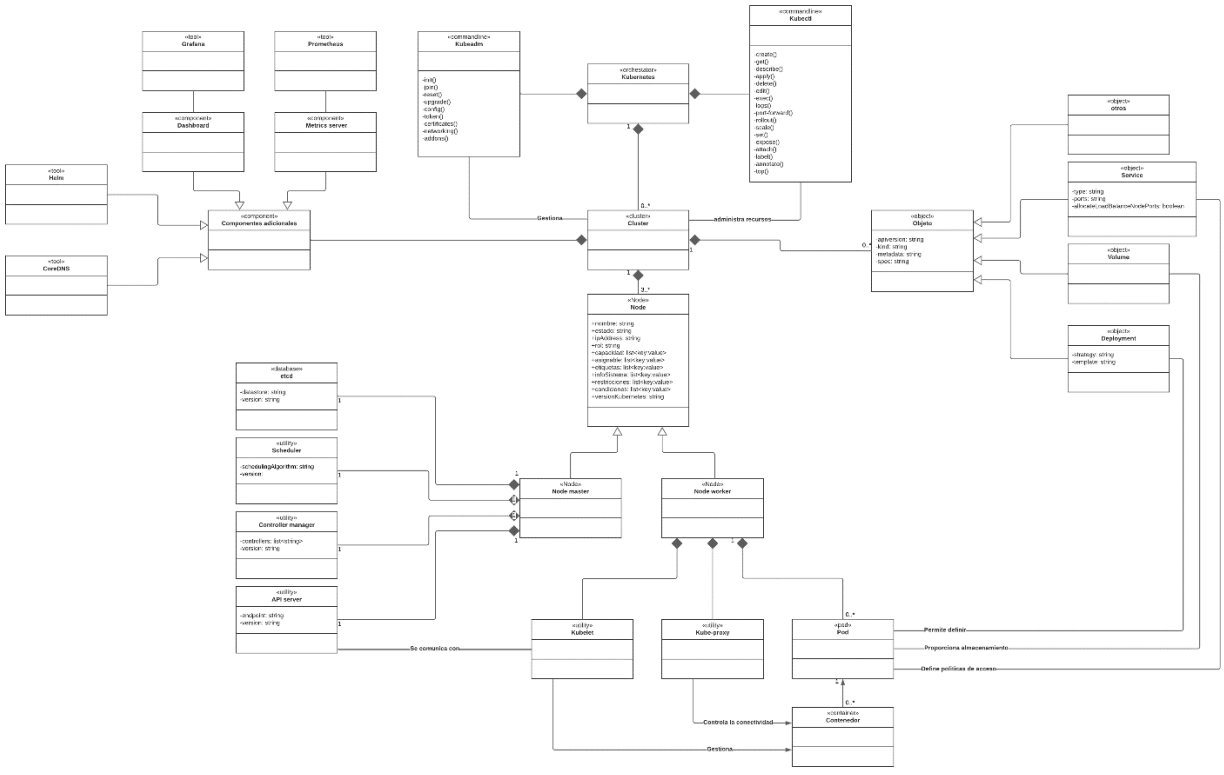

Figura 1. Metamodelo de Kubernetes

- **Clúster:** Conjunto de nodos (máquinas físicas o virtuales) que funcionan en conjunto para ejecutar aplicaciones en contenedores.

- **Nodo:** Máquina física o virtual que forma parte de un clúster de Kubernetes y es responsable de ejecutar las aplicaciones en contenedores. Como mínimo, el clúster debe tener un nodo máster y dos nodos *worker*.

- **Nodo máster:** Responsable de gestionar el estado del clúster, orquestar la carga de trabajo y realizar tareas administrativas

- **Nodo *worker*:** Ejecutan las aplicaciones en contenedores.

- **Scheduler:** Asigna los pods a los nodos en el clúster.

- **API server:** Proporciona la interfaz RESTful para interactuar con el clúster.

- **Pod:** Es una abstracción que representa una o más instancias de un contenedor (o, en algunos casos, múltiples contenedores que comparten el mismo espacio de red y almacenamiento).

- **Contenedor:** Unidad de software ligera y portátil que encapsula una aplicación y todas sus dependencias (bibliotecas, binarios, archivos de configuración, etc.) en un entorno aislado.

- **Objeto:** Artefactos que se pueden definir en un despliegue de Kubernetes dentro de un clúster.

- **Service:** Permiten la comunicación entre los diferentes componentes de una aplicación, tanto dentro como fuera del clúster de Kubernetes.

- **Deployment:** Abstracción que define un conjunto de pods que realizan la misma función y proporciona una forma de acceder a ellos de manera uniforme a través de una dirección IP y un nombre de servicio.

- **Grafana:** Un panel configurable para visualizar los datos obtenidos por Prometheus.

- **Prometheus:** Herramienta que recoge datos del clúster y los sirve en una API.

1.2 Líneas de productos software.

Una línea de productos software (SPL del inglés *Software Product Line*) es una familia de productos software relacionados entre sí que comparten ciertas características comunes (similitudes) pero que también tienen características variables. En un SPL, los sistemas se descomponen en características (*features*) que representan funcionalidades o comportamientos del sistema.

El principal artefacto para modelar la variabilidad en SPL son los modelos de variabilidad. Existen muchos modelos de variabilidad, pero los más extendidos y usados en la práctica son los modelos de características (*feature models*). Un *feature model* (FM) es un modelo para representar la variabilidad de una SPL en base a características comunes y variables (véase Figura 2), especificando qué características se pueden seleccionar en una configuración para generar un producto válido de la SPL.

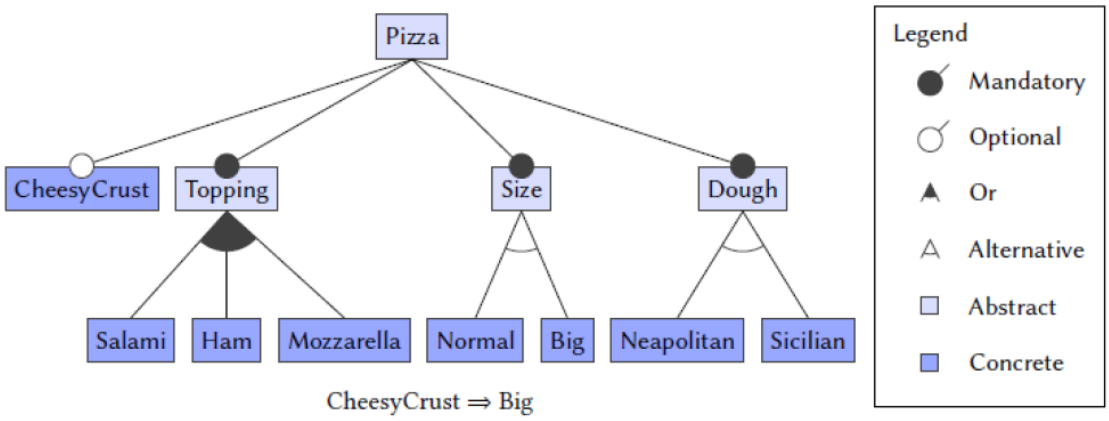

Figura 2. Feature model especificado con la herramienta FeatureIDE (*Knüppel et al., 2017*)

Un FM contiene dos partes bien diferenciadas: el árbol de características y las restricciones textuales.

1. El árbol de características (feature tree) que descompone las características en una estructura jerárquica en forma de árbol. En el árbol, existen dos tipos de características: características concretas (concrete) y características abstractas (abstract).

2. Las restricciones textuales (cross-tree constraints) son relaciones más complejas entre las características que no pueden ser modeladas en la jerarquía del árbol. Las restricciones textuales suelen especificarse con fórmulas lógicas de diferente complejidad (e.g., fórmulas proposicionales, lógica de primer orden, etc.); pero los tipos de restricciones más simples incluyen restricciones del tipo *"A requires B"* (también escrita como A => B).

Los FMs se pueden formalizar a lógica proposicional para analizar y razonar automáticamente sobre sus propiedades. Por ejemplo, para conocer si el modelo es válido, esto es, si representa al menos un producto; para generar todas configuraciones y productos válidos; para extraer aquellas características que están presentes en todas las configuraciones (conocidas como *core features*), o aquellas que no están presentes en ningún producto (conocidas como *dead features*), etc.

Una configuración válida de un modelo es un conjunto de características que cumple con todas las relaciones del árbol y restricciones textuales del modelo. Por ejemplo, una de las configuraciones del FM de ejemplo sería:

[Pizza, Topping, Mozzarella, Ham, Size, Normal, Dough, Sicilian]

La diferencia entre una configuración y un producto viene dada por la diferencia entre las características abstractas y concretas. Un producto es una configuración válida del modelo donde las características abstractas han sido eliminadas. Así, el producto resultante para la configuración del ejemplo anterior sería:

[Mozzarella, Ham, Normal, Sicilian]

Este informe se centra concretamente en los modelos de características, usando un FM de Kubernetes (anexo A) para obtener el número de configuración posibles de un fichero de configuración y para ayudar con la creación de un objeto de Kubernetes.

1.3 Entorno de trabajo y herramientas usadas

Todo el trabajo realizado que da lugar a este informe se ha hecho en el entorno de programación Visual Studio Code (a partir de ahora, VS Code), usando el lenguaje Python en gran parte, aunque usando también otros como UVL (*Universal Variability Language*) (Sundermann et al., 2021) para definir los modelos de características (*Feature Models*) o Jinja2 para la creación de las plantillas usadas en la generación de ficheros de configuración de Kubernetes.

Algunos plugin de VS Code han resultado ser muy útiles en el desarrollo. A continuación, se procede a describirlos:

- **UVLS – Universal Variability Language Server** (Loth et al., 2023)**:** Es una herramienta que ayuda durante la creación de los FM detectando errores en su definición mientras se desarrolla. Necesita la instalación de la librería llamada z3 para tener todas sus funcionalidades activas, que son la creación de una configuración del FM respetando las restricciones dadas y generar un archivo usado para visualizar el FM.

- **Graphviz interactive Preview:** Este plugin hace uso del archivo generado por el anterior plugin para mostrar el FM en una nueva ventana del VS Code. Deja de ser tan útil cuando se manejan FM con gran cantidad de características, pues visualizarlos se hace complicado debido a su tamaño.

Otra herramienta usada para el análisis de variabilidad del FM de Kubernetes es Flamapy [1](A. Galindo et al., 2023). Con ella, se han obtenido datos del modelo tales como el número de características, numero de restricciones, la media de características que se escogen en la configuración del FM, etc.

[1] Página oficial de Flamapy: https://flamapy.github.io/

2. Metodología

2.1 Generación de los ficheros de configuración

Uno de los principales objetivos que se busca conseguir es poder generar automáticamente todos los ficheros necesarios para el despliegue de una aplicación en Kubernetes de una forma intuitiva y sin necesidad de conocer todas las dependencias que existen en un proyecto. Para ello se va a usar un FM que contienen las posibles configuraciones, de forma que el usuario solo tenga que ir especificando aquellas que necesite.

Hasta el momento de escribir este informe, se han desarrollado dos versiones del FM de Kubernetes. La primera versión permite crear manifiestos YAML para creación de objetos del tipo *Deployment* (que incluyen pods y contenedores) y *Service*, Aunque de forma limitada, pues no incluye aun todas las características que se pueden definir en un manifiesto. Para la segunda versión generada de manera automática se realizó un proceso distinto en el punto 3. Para poder generar el fichero de configuración, se requieren de 4 elementos:

El *feature model* del que se consigue una configuración válida en formato JSON, una plantilla definida con Jinja2 y un archivo csv que contiene una tabla que relaciona los elementos de la plantilla con los valores de la configuración.

1. ***Feature Model*:** Aquellas características que se quieran reflejar en la plantilla se indican en la primera columna de la tabla de mapeo, en la columna de *features*.

2. **Configuración válida:** A partir del FM se consigue un archivo JSON que contiene los valores de las características seleccionados para esta configuración.

3. **Plantilla:** Una plantilla hecha con Jinja2 que contiene referencias a los valores almacenados en el JSON de la configuración válida, y que define la estructura final del archivo de configuración.

4. **Archivo de mapeo:** Un archivo csv que contiene una tabla con 3 columnas; *Feature*, *Handler* y *Value*. La primera es el nombre de la *característica* en el archivo de configuración, de la que se consulta su valor. La segunda es el nombre de la variable en la plantilla donde se pondrá el valor referenciado en la primera columna.

Una vez se disponen de estos ficheros, con ayuda de un script en Python basado en el código del GitHub uvengine (Jmhorcas. (s. f.)) y cuyo código, con ligeras modificaciones, se puede encontrar en GitHub fms_dataset (CAOSD-group. (s. f.).) se obtiene el fichero de configuración listo para desplegarlo en Kubernetes.

Como paso adicional, se pueden usar herramientas de validación como Kubeconform (Yannh. (s. f.)) o Kube-score [2](Zegl. (s. f.)) en la ejecución de este script para comprobar que el fichero resultante está bien definido, eliminando la necesidad de tener que desplegarlo en un entorno real para descubrir errores. En el anexo B se da más información acerca de la ejecución del script añadiendo la validación con estas herramientas.

[2] Herramienta online de Kube-score: https://kube-score.com/

2.2 Resultados del análisis de variabilidad del FM de Kubernetes

Usando la herramienta Flamapy ya mencionada anteriormente, se ha conseguido analizar la variabilidad del FM de Kubernetes, obteniendo los siguientes datos en esta primera versión:

Con el modelo actual, los datos obtenidos nos dicen que se suelen tener 186 atributos de media en una configuración, siendo la configuración mínima de tan solo 9 características, y la máxima de 282. En la figura 3 se observa mejor la distribución, agrupándose la mayoría de las configuraciones entre 150 y 220 características aproximadamente.

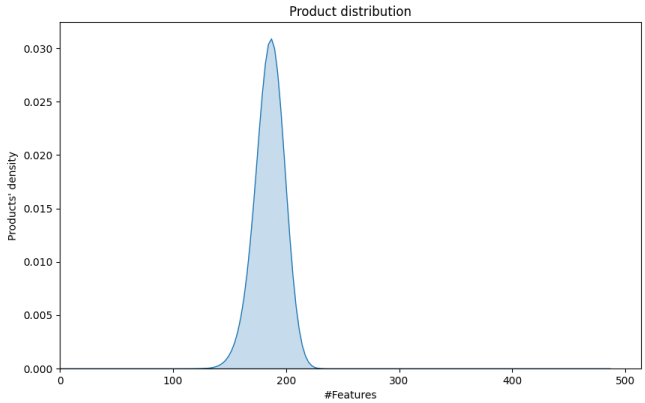

Figura 3: Gráfico de distribución de las características del FM de Kubernetes.

La figura 4 proporciona algo más de información sobre el uso de las características, entendiendo que a más a la derecha se encuentre, más probable es que esa característica se incluya en una configuración valida. Por ejemplo, vemos a la izquierda un número importante de ellas que en muy pocas ocasiones se incluyen en una configuración valida, aunque no llegan a ser características a las que nunca se acceden (*dead features*).

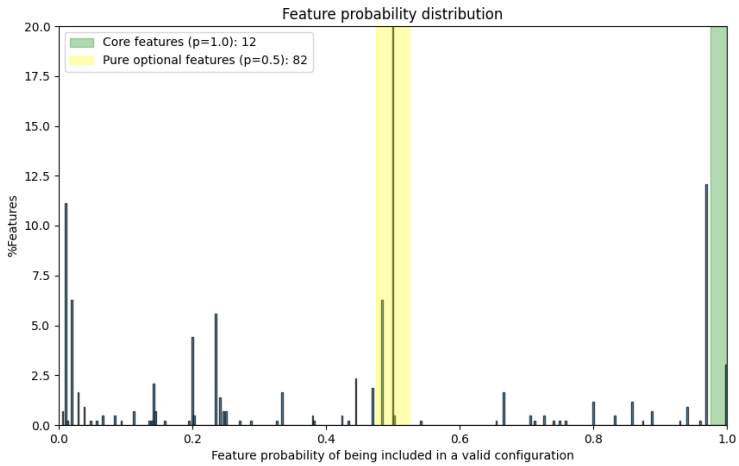

Figura 4: Distribución de probabilidad de una característica de ser incluida en una configuración valida del FM de Kubernetes.

En definitiva, este FM de Kubernetes tiene 431 características y 43 restricciones, elevando el número de configuraciones posibles a 6.78e42 configuraciones validas aproximadamente.

3. Generación automática del modelo de variablidad de Kubernetes

En este punto se exponen detalles de un modelo generado de manera automática. Se procede a explicar brevemente la generación automática de los *features* y las *constraints* a partir de los esquemas JSON de Kubernetes. Por último, se exponen el número de features, restricciones y configuraciones obtenidas.

3.1 Generación automática de las características

En este apartado se introduce información sobre el modelo generado automáticamente mediante la ejecución de un programa en Python. Este programa procesa los esquemas en formato JSON y los mapea automáticamente a un modelo de características uvl. Los ficheros que se han usado se muestran en el repositorio de la versión 1.30.2 [3], concretamente se ha usado el archivo *_definitions* para realizar el mapeo de JSON a características, ya que en ese archivo se encuentra el contenido de los demás archivos de forma más compacta y ordenada.

El programa que realiza el mapeo se encuentra en GitHub [4]. Este script se encarga de procesar todos los esquemas JSON y generar el FM completo [5], estas son parte de las funciones que realiza:

- **Mapear los nombres de los features**. Se mantiene el nombre original de los esquemas modificando los caracteres no válidos en uvl. Para representar las propiedades que contienen los esquemas se encadenan los nombres de dichas propiedades al nombre del padre. En la Figura 5 se puede observar el esquema llamado *"io.k8s.api.admissionregistration.v1.AuditAnnotation",* este nombre se convierte a "*io_k8s_api_admissionregistration_v1_AuditAnnotation"* en el modelo. Para agregar las propiedades (hijos o características) de los esquemas se concadena al padre: *io_k8s_api_admissionregistration_v1_AuditAnnotation_key, io_k8s_api_admissionregistration_v1_AuditAnnotation_valueExpression*. Siguiendo el nivel que se recorre y sin perder el nombre de los esquemas anteriores se forman los sub-features. En la Figura 6 se puede observar el esquema mapeado.

[3] Esquemas de la versión de Kubernetes 1.30.2: https://github.com/yannh/kubernetes-json-schema/tree/master/v1.30.2
[4] Script que genera el feature model uvl: https://github.com/CAOSD-group/fm-json-kubernetes/blob/main/scriptJsonToUvl/convert01.py
[5] Modelo final: https://github.com/CAOSD-group/fm-json-kubernetes/blob/main/scriptJsonToUvl/kubernetes_combined_01.uvl

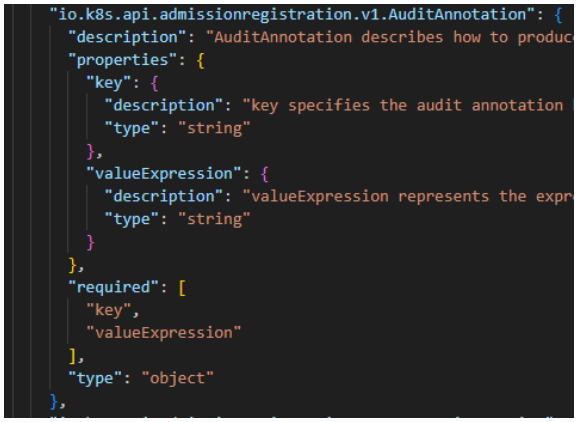

Figura 5: Esquema JSON de ejemplo AuditAnnotation

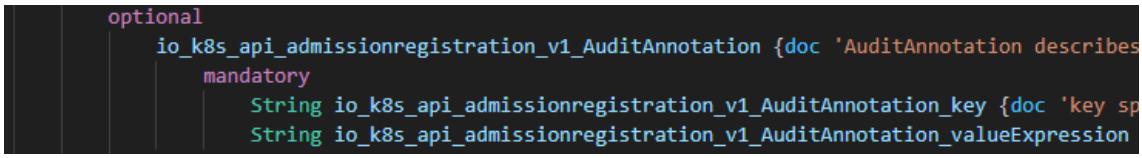

Figura 6: Feature AuditAnnotation en el modelo uvl

- Mantener la estructura/jerarquía La jerarquía del modelo. En base a los esquemas JSON que ya disponen de una jerarquía base, se sabe que, de un nivel superior, los features se van expandiendo en un mismo nivel o hacia abajo. Se recorre el archivo de manera ordenada y se extiende según se encuentren anidamientos o nuevos esquemas. Para completar la jerarquía se hizo uso de la característica "*required*" para definir si un feature era obligatorio o no. Dando por hecho, que las propiedades que no se mencionen en dicha lista son opcionales. Como se puede observar en la Figura 6, al estar las dos propiedades dentro de *required* se agregaron en el modelo bajo *mandatory* (obligación de que aparezca).

- Completar las referencias. Las referencias son enlaces a otros esquemas que se usan para agregar los esquemas referenciados al esquema principal. De esta manera se reutilizan esquemas y no hay necesidad de volver a representar el esquema al completo. En la Figura 7 se puede observar el esquema *MatchResources* que contiene una referencia, precedida por *"$ref"*, en "*excludeResourceRules*". Esta referencia se muestra en el modelo de manera completa, extendiendo la propiedad con el contenido del esquema referenciado. Por lo tanto, en vez de usar referencias entre features se han extendido por completo las referencias.

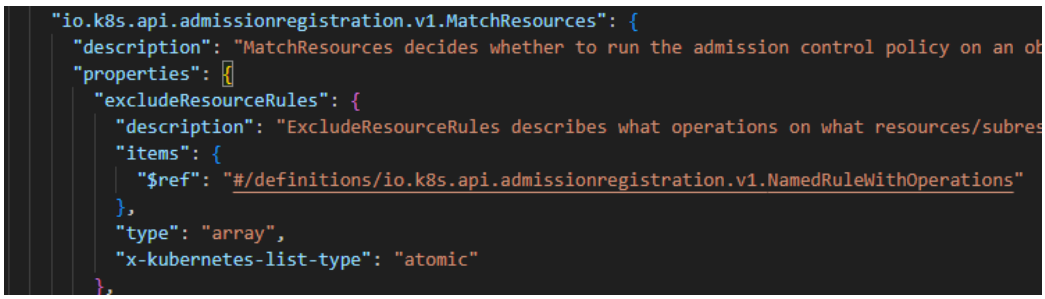

Figura 7: Esquema JSON con una referencia en la propiedad excludeResourceRules

- **Usar las descripciones de los esquemas para completar el modelo.** Las descripciones se localizan bajo la característica *description* y contiene información sobre el esquema o propiedad que define. De estas descripciones se generan valores de los features inhibidos en el texto y se usan palabras clave para modificar la jerarquía. En algunas descripciones se insertan valores que pueden tomar los features y mediante patrones de coincidencia se extraen dichos valores para representarlos como sub-features. Por otro lado, también se insertan palabras al final de las descripciones como "*Required.*", que en algunos casos coincide con el contenido de la característica *required* vista en el segundo punto, pero en otros no. En los casos de no coincidencia se modifica el feature como obligatorio.

3.2 Generación automática de las restricciones

Después de completar el modelo se procedió a generar restricciones de manera automática. El procedimiento se basó en buscar patrones de texto que se repitiesen y tuvieran una función de dependencia, restricción o intervalo entre las características. Es decir, se buscaron palabras clave para crear grupos de restricciones entre los features. Este proceso se realiza en el script [6] situado en el repositorio anterior. De forma general el script usa las descripciones filtradas en el archivo *descriptions_01.json* [7] y analiza cada una buscando coincidencias con las funciones que se implementan en el script de Python. En total se crean de manera totalmente automática 5.364 restricciones de varios tipos. Se podrá ver en detalle más adelante.

Se puede observar también la siguiente tabla que resume las funciones del script, expresiones que se usan, una pequeña descripción de la restricción que trata de abarcar y el número que precede cada una:

Función	Expresiones	Descripción	Nº
extract_constraints _os_name()	re.compile(r'(?<=Note that this field cannot be set when spec.os.name is\s)([a-zA-Z\s,]+)(?=\.)', re.IGNORECASE)	Se define el posible uso de un feature o no en base al sistema operativo que se seleccione. Dependiendo del S.O se habilitan o deshabilitan features no compatibles.	1247
extractBounds()	re.compile(r'(\d+)\s*<\s*\w+\s*<\s*(\d+)') re.compile(r'(\d+)\s*-\s*(\d+)\s*\(inclusive\)') re.compile(r'Number\s+must\s+be\s+in\s+ the\s+range\s+(\d+)\s+to\s+(\d+)'	Agrupación de constraints de intervalos y limites maximos y minimos. Se manejan las descripciones que contienen rangos o	>700

[6] Script para analizar las descripciones y generar restricciones: https://github.com/CAOSD-group/fm-json-kubernetes/blob/main/scriptJsonToUvl/analisisScriptNpl01.py
[7] Archivo con las descripciones filtradas: https://github.com/CAOSD-group/fm-json-kubernetes/blob/main/scriptJsonToUvl/descriptions_01.json

	re.compile(r'must\s+be\s+between\s+(\d+)\s+and\s+(\d+)'	límites de minimos y/o máximos en puertos.	
extractBounds()	re.compile(r'must be greater than(?: or equal to)? (\w+)' re.compile(r'less than or equal to (\d+)', re.IGNORECASE)	Agrupación de restricciones valores que por defecto son mayor a cero y rangos en segundos. Se tratan las descripciones que tienen el valor del mínimo como número entero o como nombre, en algunos casos se representaba el 0 como "zero" y se realizó una conversión entre esos valores para obtener un valor entero.	>400
extract_constraints_mutualy_exclusive()	re.compile(r'\`([A-Za-z]+)\`')	Se basa en la exclusión mutua entre 2 propiedades. Si una es seleccionada la otra no debe de estarlo.	12
extract_constraints_if()	re.compile(r'\"([A-Za-z]+)\"')	Se basa en al valor que pueda tomar la propiedad "type", el cual debe ser escogido para que otra propiedad deba de ser configurada. La otra propiedad o localhostProfile es el sub-feature que depende del valor que tome "type". Solo si es el valor indicado en la descripción será configurado y no otro.	>300
extract_constraints_required_when()	re.compile(r'Required when\s+(\w+)\s+is\s+set\s+to\s+"([^"]+)"', re.IGNORECASE) re.compile(r'must be unset when\s+(\w+)\s+is\s+set\s+to\s+"([^"]+)"', re.IGNORECASE) re.compile(r'Required when\s+`(\w+)`\s+is\s+set\s+to\s+`?\"?([^\"`]+)\"?`?', re.IGNORECASE)	Se basa en el uso un valor de una propiedad que, si se usa, otra propiedad debe de ser seleccionada. El tratamiento en general es similar a la anterior agrupación, pero varía en cuanto a los términos y la implementación.	8

extract_constraints_operator()	re.compile(r'If the operator is\s+(\w+)\s+or\s+(\w+)', re.IGNORECASE) re.compile(r'If the operator is\s+(\w+)')	Se basa en varios casos, en muchos hay pares de valores que si se seleccionan encadenan a otro feature a que se seleccione. Por otro lado, hay un caso aislado donde se menciona solo un posible valor para realizar la dependencia.	701
extract_constraints_least_one()	re.compile(r'(?<=a least one of\s)(\w+)\s+or\s+(\w+)', re.IGNORECASE) re.compile(r'(?<=Exactly one of\s)`(\w+)`\s+or\s+`(\w+)`', re.IGNORECASE) re.compile(r'(?<=At least one of\s)`(\w+)`\s+and\s+`(\w+)`', re.IGNORECASE)	Se asegura de que al menos uno de los features deba de ser seleccionado. Precedido por las expresiones one of.	>450
extract_constraints_primary_or()	Sin patrones, basado en palabras clave de las descripciones.	Se basa en obtener relaciones entre features que no tienen descritas su relación, pero si en un nivel más arriba. En su mayoría las restricciones suelen ser del tipo: "feature1 or feature2".	56
extract_constraints_multiple_conditions()	re.compile(r'\b(Approved|Denied|Failed)\b') re.compile(r'(?<=conditions may not be\s)\"([A-Za-z]+)\"\s+or\s+\"([A-Za-z]+)\"')	Se agregan las restric complejas, son largas y contienen varios operadores lógicos o features involucrados. No tienen relación entre todas, pero se diferencian por la longitud de estas.	25
extract_constraints_string_oneOf()	Sin patrones, basado en palabras clave de las descripciones.	Se tratan restricciones de tipo Sting, en ellas: La primera obtiene restricciones en base al tipo del tipo de petición que defina el campo kind (String). Al no tener definido los tipos en su	10

		descripción no se los ha insertado como Booleans pero si se usa para aplicar la coincidencia y activación de los otros campos	
extract_constraints_template_onlyAllowed	re.compile(r'(?<=The only allowed template.spec.restartPolicy value is\s)\"([A-Za-z]+)\"', re.IGNORECASE) re.compile(r'\"([A-Za-z]+)\"')	Se basan en el uso de la expresión *template.spec.restartPolicy* para capturar el valor permitido en ese esquema o feature. Dependiendo del valor indicado se permiten el/los obtenidos y se quitan los demás. Relacionado con las políticas de reinicio.	19
extract_minimum_value()	re.compile(r'(?<=Minimum value is\s)(\d+)') re.compile(r'(?<=minimum valid value for expirationSeconds is\s)(\d+)') re.compile(r'(?<=in the range\s)(\d+)-(\d+)') “Value must be non-negative”	Se utilizan las expresiones para capturar los valores enteros mínimos de límites en features. Se usan segundos, nº repeticiones. También se agrega un rango con límites 1-100.	1.426
Nº total de constraints:		**5.364**	

3.3 Resultados del análisis de variabilidad del FM de Kubernetes

En este apartado se explica el procedimiento para tratar de analizar el modelo final obtenido. Debido a la magnitud del modelo no se pudo realizar un procedimiento similar al anterior con flamapy. Por ello, se usó una herramienta similar llamada fmfactlabel [8] con algunas modificaciones. Mediante la versión *light,* Figura 8*,* se pudo lograr analizar y obtener una estimación del número de configuraciones resultantes, además de otros parámetros como el número de features en total y distintos tipos de restricciones. En este caso se muestran los más interesantes que son los datos del número de características, las restricciones y una aproximación de las configuraciones posibles.

[8] Herramienta usada para el análisis: https://fmfactlabel.adabyron.uma.es/?v=1.7.0

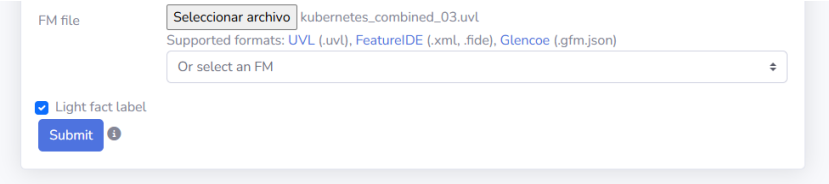

Figura 8: Opción light en la herramienta factlabel

Resaltando los números obtenidos, son un total de 63.737 features, 5.364 restricciones y una aproximación de 9.44 x 10 elevado a 10.263 configuraciones. Este último dato es una aproximación debido a que actualmente no se puede procesar y calcular el número de configuraciones de un modelo tan grande. El resultado del análisis se puede observar en la figura siguiente.

Features	63737	
Abstract features	825	(1%)
Abstract compound features	825	(100%)
Concrete features	62912	(99%)
Concrete leaf features	42952	(68%)
Concrete compound features	19960	(32%)
Compound features	20785	(33%)
Leaf features	42952	(67%)
Root feature	1	(0%)
Top features	626	(1%)
Solitary features	51986	(82%)
Grouped features	11716	(18%)
Typed features	29593	(46%)
Numerical features	6066	(10%)
Integer features	6066	(10%)
String features	23527	(37%)
Multi-features	6455	(10%)
Tree relationships	56235	
Mandatory features	14208	(27%)
Optional features	37812	(73%)
Feature groups	4215	(7%)
Alternative groups	4215	(100%)
Depth of tree	15	
Mean depth of tree	6.77	
Branching factor	3.07	
Min children per feature	1	
Max children per feature	626	
Avg children per feature	1	
Cross-tree constraints	5364	
Logical constraints	2547	(47%)
Simple constraints	1293	(51%)
Requires constraints	46	(4%)
Excludes constraints	1247	(96%)
Complex constraints	1254	(49%)
Pseudo-complex constraints	32	(3%)
Strict-complex constraints	1222	(97%)
Arithmetic constraints	2817	(53%)
Features in constraints	11150	(17%)
Min features per constraint	1	
Max features per constraint	7	
Avg features per constraint	2.37	
Avg constraints per feature	1.14	
Min constraints per feature	1	
Max constraints per feature	78	
Attributes	1	
Features with attributes	3656	(6%)
Max attributes per feature	1	
Avg attributes per feature	0.06	
Avg attributes per feature w. attributes	1	
Satisfiable (valid)	Yes	
Core features	1	(0%)
False-optional features	579	(1%)
Dead features	914	(1%)
Variant features	62822	(99%)
Configurations	≤ 9.44e10263	
Total variability	0.0%	
Partial variability	0.0%	

Figura 9: Análisis con detalles del modelo generado automáticamente

4. Conclusiones

Del desarrollo de este proyecto se concluye que es posible afrontar el problema de la configuración de un despliegue de Kubernetes con un enfoque de líneas de productos software y más concretamente de modelos de características. Se ha conseguido desarrollar una primera versión del FM de Kubernetes para su análisis de variabilidad, dando como resultado un modelo de un tamaño muy elevado y con un gran número de características usadas frecuentemente. Estos resultados podrían aplicarse a otras tecnologías (como Docker) que requieran de grandes configuraciones que no son triviales para el usuario, aportando la idea de una nueva herramienta que agiliza el trabajo de desarrolladores e investigadores.

Referencias

1. Alexander Knüppel, Thomas Thüm, Stephan Mennicke, Jens Meinicke, and Ina Schaefer. 2017. Is there a mismatch between real-world feature models and product-line research? In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017). Association for Computing Machinery, New York, NY, USA, 291–302. https://doi.org/10.1145/3106237.3106252.

2. Chico Sundermann, Kevin Feichtinger, Dominik Engelhardt, Rick Rabiser, and Thomas Thüm. 2021. Yet another textual variability language? a community effort towards a unified language. In Proceedings of the 25th ACM International Systems and Software Product Line Conference - Volume A (SPLC '21). Association for Computing Machinery, New York, NY, USA, 136–147. https://doi.org/10.1145/3461001.3471145.

3. Jacob Loth, Chico Sundermann, Tobias Schrull, Thilo Brugger, Felix Rieg, and Thomas Thüm. 2023. UVLS: A Language Server Protocol For UVL. In Proceedings of the 27th ACM International Systems and Software Product Line Conference - Volume B (SPLC '23), Vol. B. Association for Computing Machinery, New York, NY, USA, 43–46. https://doi.org/10.1145/3579028.3609014.

4. José A. Galindo, Jose-Miguel Horcas, Alexander Felferning, David Fernandez-Amoros, and David Benavides. 2023. FLAMA: A collaborative effort to build a new framework for the automated analysis of feature models. In Proceedings of the 27th ACM International Systems and Software Product Line Conference - Volume B (SPLC '23), Vol. B. Association for Computing Machinery, New York, NY, USA, 16–19. https://doi.org/10.1145/3579028.3609008.

5. Jmhorcas. (s. f.). GitHub - jmhorcas/uvengine: SPL implementation of the variability using templates. GitHub. https://github.com/jmhorcas/uvengine.

6. CAOSD-group. (s. f.). GitHub - CAOSD-group/fms_dataset: Dataset of feature models of the CAOSD research group. GitHub. https://github.com/CAOSD-group/fms_dataset.

7. Yannh. (s. f.). GitHub - yannh/kubeconform: A FAST Kubernetes manifests validator, with support for Custom Resources! GitHub. https://github.com/yannh/kubeconform.

8. Zegl. (s. f.). GitHub - zegl/kube-score: Kubernetes object analysis with recommendations for improved reliability and security. kube-score actively prevents downtime and bugs in your Kubernetes YAML and Charts. Static code analysis for Kubernetes. GitHub. https://github.com/zegl/kube-score

Anexo A

Modelo de características de Kubernetes

```
features
    Kubernetes_manifest
        mandatory
            Boolean ApiVersion
                mandatory
                    Boolean Group {abstract}
                        alternative
                            Boolean GROUP_admission_registration
                            Boolean GROUP_apiextensions
                            Boolean GROUP_apiregistration
                            Boolean GROUP_apps
                            Boolean GROUP_authentication
                            Boolean GROUP_authorization
                            Boolean GROUP_autoscaling
                            Boolean GROUP_batch
                            Boolean GROUP_certificates
                            Boolean GROUP_coordination
                            Boolean GROUP_core
                            Boolean GROUP_discovery
                            Boolean GROUP_events
                            Boolean GROUP_flowcontrol_apiserver
                            Boolean GROUP_internal_apiserver
                            Boolean GROUP_networking
                            Boolean GROUP_node
                            Boolean GROUP_policy
                            Boolean GROUP_rbac_authorization
                            Boolean GROUP_resource
                            Boolean GROUP_scheduling
                            Boolean GROUP_storage
                    Boolean Version {abstract}
                        alternative
                            Boolean VERSION_v1
                            Boolean VERSION_v1beta1
                            Boolean VERSION_v1beta3
                            Boolean VERSION_v1alpha1
                            Boolean VERSION_v1alpha2
                            Boolean VERSION_v2
```

```
Boolean Kind
    alternative
        Boolean Workloads_APIs {abstract}
            alternative
                Boolean KIND_Container
                Boolean KIND_CronJob
                Boolean KIND_DaemonSet
                Boolean KIND_Deployment
                Boolean KIND_Job
                Boolean KIND_Pod
                Boolean KIND_ReplicaSet
                Boolean KIND_ReplicationController
                Boolean KIND_StatefullSet
        Boolean Service_APIs {abstract}
            alternative
                Boolean KIND_Endpoints
                Boolean KIND_EndpintSlice
                Boolean KIND_Ingress
                Boolean KIND_IngressClass
                Boolean KIND_Service
        Boolean Confing_and_storage_APIs {abstract}
            alternative
                Boolean KIND_ConfigMap
                Boolean KIND_CSIDriver
                Boolean KIND_CSINode
                Boolean KIND_Secret
                Boolean KIND_PersistentVolumeClaim
                Boolean KIND_StorageClass
                Boolean KIND_CSIStorageCapacity
                Boolean KIND_Volume
                Boolean KIND_VolumeAttachment
                Boolean KIND_VolumeAttributesClass
        Boolean Metadata_APIs {abstract}
            alternative
                Boolean KIND_ClusterTrustBundle
                Boolean KIND_ControllerRevision
                Boolean KIND_CustomerResourceDefinition
                Boolean KIND_Event
                Boolean KIND_LimitRange
                Boolean KIND_HorizontalPodAutoscaler
                Boolean KIND_MutatingWebhookConfiguration
                Boolean KIND_ValidatingWebhookConfiguration
```

```
                    Boolean KIND_PodSchedulingContext
                    Boolean KIND_PodTemplate
                    Boolean KIND_PodDisruptionBudget
                    Boolean KIND_PriorityClass
                    Boolean KIND_ResourceClaim
                    Boolean KIND_ResourceClaimParameters {added_version 1.30}
                    Boolean KIND_ResourceSlice {added_version 1.30}
                    Boolean KIND_ResourceClaimTemplate
                    Boolean KIND_ResourceClass
                    Boolean KIND_ValidatingAdmissionPolicy
                    Boolean KIND_ValidatingAdmissionPolicyBinding
                Boolean Cluster_APIs {abstract}
                    alternative
                        Boolean KIND_APIService
                        Boolean KIND_Binding
                        Boolean KIND_CertificateSigningRequest
                        Boolean KIND_ClusterRole
                        Boolean KIND_ClusterRoleBinding
                        Boolean KIND_ComponentStatus
                        Boolean KIND_FlowSchema
                        Boolean KIND_IPAddress
                        Boolean KIND_Lease
                        Boolean KIND_LocalSubjectAccessReview
                        Boolean KIND_Namespace
                        Boolean KIND_Node
                        Boolean KIND_PersistentVolume
                        Boolean KIND_PriorityLevelConfiguration
                        Boolean KIND_ResourceQuota
                        Boolean KIND_Role
                        Boolean KIND_RoleBinding
                        Boolean KIND_RuntimeClass
                        Boolean KIND_SelfSubjectAccessReview
                        Boolean KIND_SelfSubjectReview
                        Boolean KIND_SelfSubjectRulesReview
                        Boolean KIND_ServiceAccount
                        Boolean KIND_ServiceCIDR
                        Boolean KIND_StorageVersion
                        Boolean KIND_SubjectAccessReview
                        Boolean KIND_TokenRequest
                        Boolean KIND_TokenReview
                        Boolean KIND_NetworkPolicy
optional
```

```
Boolean Metadata
    mandatory
        String METADATA_name
    optional
        String METADATA_namespace
        Boolean METADATA_labels cardinality [0..*]
            mandatory
                String LABELS_key
                String LABELS_value
        Boolean METADATA_annotations cardinality [0..*]
            mandatory
                String ANNOTATIONS_key
                String ANNOTATIONS_value
Boolean Spec
    optional
        Boolean DeploymentSpec {abstract}
            optional
                Integer DEPLOYMENTSPEC_minReadySeconds
                Boolean DEPLOYMENTSPEC_paused
                Integer DEPLOYMENTSPEC_progressDeadlineSeconds
                Integer DEPLOYMENTSPEC_Replicas
                Integer DEPLOYMENTSPEC_revisionHistoryLimit
                Boolean DEPLOYMENTSPEC_LabelSelector
                    or
                        Boolean LABELSELECTOR_matchLabels cardinality [0..*]
                            mandatory
                                String MATCHLABELS_key
                                String MATCHLABELS_value
                        Boolean LABELSELECTOR_matchExpressions cardinality [0..*]
                            mandatory
                                String MATCHEXPRESSIONS_key
                                Boolean MATCHEXPRESSIONS_values cardinality [0..*]
                                    mandatory
                                        String MATCHEXPRESSIONS_value
                                Boolean MATCHEXPRESSIONS_operator
                                    alternative
                                        Boolean OPERATOR_in
                                        Boolean OPERATOR_notIn
                                        Boolean OPERATOR_exist
                                        Boolean OPERATOR_doesNotExist
                                        Boolean OPERATOR_gt
                                        Boolean OPERATOR_lt
```

```
        Boolean DEPLOYMENTSPEC_strategy
            mandatory
                Boolean STRATEGY_type
                    or
                        Boolean Recreate
                        Boolean RollingUpdate
                            mandatory
                                Integer maxUnavailable
                            optional
                                Integer maxSurge
        Boolean DEPLOYMENTSPEC_template
            mandatory
                Boolean DEPLOYMENTSPEC_TEMPLATE_Metadata
                    optional
                        String TEMPLATE_METADATA_name
                        String TEMPLATE_METADATA_namespace
                        Boolean TEMPLATE_METADATA_labels cardinality [0..*]
                            mandatory
                                String TEMPLATE_METADATA_LABELS_key
                                String TEMPLATE_METADATA_LABELS_value
                        Boolean TEMPLATE_METADATA_annotations cardinality [0..*]
                            mandatory
                                String TEMPLATE_ANNOTATIONS_key
                                String TEMPLATE_ANNOTATIONS_value
                Boolean DEPLOYMENTSPEC_TEMPLATE_spec
Boolean PodSpec {abstract}
    optional
        Integer PODSPEC_activeDeadlineSeconds
        Boolean PODSPEC_affinity
            optional
                Boolean AFFINITY_nodeAffinity
                Boolean AFFINITY_podAffinity
                Boolean AFFINITY_podAntiAffinity
        Boolean PODSPEC_automountServiceAccountToken
        Boolean PODSPEC_Containers cardinality [0..*]
            optional
                Boolean CONTAINER_args cardinality [0..*]
                    mandatory
                        String CONTAINER_ARGS_value
                Boolean CONTAINER_command cardinality [0..*]
                    mandatory
                        String CONTAINER_COMMAND_value
```

```
Boolean CONTAINER_env cardinality [0..*]
    mandatory
        String CONTAINER_ENV_name
        String CONTAINER_ENV_value
        Boolean CONTAINER_ENV_valueFrom
            alternative
                Boolean configMapKeyRef
                    mandatory
                        String CONFIGMAPKEYREF_key
                        String CONFIGMAPKEYREF_name
                        Boolean CONFIGMAPKEYREF_optional
                Boolean fieldRef
                    mandatory
                        String FIELDREF_apiversion
                        String FIELDREF_fieldPath
                Boolean resourceFieldRef
                    mandatory
                        String RESOURCFIELDREF_containerName
                        String RESOURCFIELDREF_divisor
                        String RESOURCFIELDREF_resource
                Boolean secretKeyRef
                    mandatory
                        String SECRETKEYREF_key
                        String SECRETKEYREF_name
                        Boolean SECRETKEYREF_optional
Boolean CONTAINER_envFrom
    mandatory
        Boolean ENVFROM_configMapRef
            mandatory
                String CONFIGMAPREF_name
                Boolean CONFIGMAPREF_optional
        String ENVFROM_prefix
        Boolean ENVFROM_secretRef
            mandatory
                String SECRETREF_name
                Boolean SECRETREF_optional
String CONTAINER_image
Boolean CONTAINER_imagePullPolicy
    alternative
        Boolean IMAGEPULLPOLICY_always
        Boolean IMAGEPULLPOLICY_never
        Boolean IMAGEPULLPOLICY_ifNotPresent
```

```
Boolean CONTAINER_lifecycle
    optional
        Boolean LIFECYCLE_postStart
            alternative
                Boolean POSTSTART_exec
                    mandatory
                        String POSTSTART_EXEC_command
                Boolean POSTSTART_httpGet
                    mandatory
                        String POSTSTART_HTTPGET_host
                        Boolean POSTSTART_HTTPGET_httpHeaders
                            mandatory
                                String POSTSTART_HTTPHEADERS_name
                                String POSTSTART_HTTPHEADERS_value
                        String POSTSTART_HTTPGET_path
                        String POSTSTART_HTTPGET_port
                        String POSTSTART_HTTPGET_scheme
                Boolean POSTSTART_sleep
                    mandatory
                        Integer POSTSTART_SLEEP_seconds
        Boolean LIFECYCLE_preStop
            alternative
                Boolean PRESTOP_exec
                    mandatory
                        String PRESTOP_EXEC_command
                Boolean PRESTOP_httpGet
                    mandatory
                        String PRESTOP_HTTPGET_host
                        Boolean PRESTOP_HTTPGET_httpHeaders
                            mandatory
                                String PRESTOP_HTTPHEADERS_name
                                String PRESTOP_HTTPHEADERS_value
                        String PRESTOP_HTTPGET_path
                        String PRESTOP_HTTPGET_port
                        String PRESTOP_HTTPGET_scheme
                Boolean PRESTOP_sleep
                    mandatory
                        Integer PRESTOP_SLEEP_seconds
Boolean CONTAINER_livenessProbe
    mandatory
        Boolean LIVENESSPROBE_exec
            mandatory
```

```
                String LIVENESSPROBE_EXEC_command
            Integer LIVENESSPROBE_failureThreshold
            Boolean LIVENESSPROBE_grpc
                mandatory
                    Integer LIVENESSPROBE_GRPC_port
                    String LIVENESSPROBE_GRPC_service
            Boolean LIVENESSPROBE_httpGet
                optional
                    String LIVENESSPROBE_HTTPGET_host
                    Boolean LIVENESSPROBE_HTTPGET_httpHeaders
                        mandatory
                            String LIVENESSPROBE_HTTPHEADERS_name
                            String LIVENESSPROBE_HTTPHEADERS_value
                    String LIVENESSPROBE_HTTPGET_path
                    Integer LIVENESSPROBE_HTTPGET_port
                    String LIVENESSPROBE_HTTPGET_scheme
            Integer LIVENESSPROBE_initialDelaySeconds
            Integer LIVENESSPROBE_periodSeconds
            Integer LIVENESSPROBE_successThreshold
            Boolean LIVENESSPROBE_tcpSocket
                mandatory
                    Boolean LIVENESSPROBE_TCPSOCKET_host
                    Boolean LIVENESSPROBE_TCPSOCKET_port
            Integer LIVENESSPROBE_terminationGracePeriodSeconds
            Integer LIVENESSPROBE_timeoutSeconds
    String CONTAINER_name
    Boolean CONTAINER_ports cardinality [0..*]
        mandatory
            Integer CONTAINER_PORTS_containerPort
        optional
            String CONTAINER_PORTS_hostIP
            Integer CONTAINER_PORTS_hostPort
            String CONTAINER_PORTS_name
            Boolean CONTAINER_PORTS_protocol
                alternative
                    Boolean PROTOCOL_UDP
                    Boolean PROTOCOL_TCP
                    Boolean PROTOCOL_SCTP
    Boolean CONTAINER_readinessProbe
        mandatory
            Boolean READINESSPROBE_exec
                mandatory
```

```
                String READINESSPROBE_EXEC_command
            Integer READINESSPROBE_failureThreshold
            Boolean READINESSPROBE_grpc
                mandatory
                    Integer READINESSPROBE_GRPC_port
                    String READINESSPROBE_GRPC_service
            Boolean READINESSPROBE_httpGet
                optional
                    String READINESSPROBE_HTTPGET_host
                    Boolean READINESSPROBE_HTTPGET_httpHeaders
                        mandatory
                            String READINESSPROBE_HTTPHEADERS_name
                            String READINESSPROBE_HTTPHEADERS_value
                    String READINESSPROBE_HTTPGET_path
                    Integer READINESSPROBE_HTTPGET_port
                    String READINESSPROBE_HTTPGET_scheme
            Integer READINESSPROBE_initialDelaySeconds
            Integer READINESSPROBE_periodSeconds
            Integer READINESSPROBE_successThreshold
            Boolean READINESSPROBE_tcpSocket
                mandatory
                    Boolean READINESSPROBE_TCPSOCKET_host
                    Boolean READINESSPROBE_TCPSOCKET_port
            Integer READINESSPROBE_terminationGracePeriodSeconds
            Integer READINESSPROBE_timeoutSeconds
    Boolean CONTAINER_resizePolicy cardinality [0..*]
        mandatory
            Boolean RESIZEPOLICY_resourceName
                alternative
                    Boolean CONTAINER_RESIZEPOLICY_RESOURCENAME_cpu
                    Boolean CONTAINER_RESIZEPOLICY_RESOURCENAME_memory
            String RESIZEPOLICY_restartPolicy
    Boolean CONTAINER_resources
        or
            Boolean RESOURCES_request
                mandatory
                    String REQUEST_cpu
                    String REQUEST_memory
            Boolean RESOURCES_limits
                mandatory
                    String LIMITS_cpu
                    String LIMITS_memory
```

```
        Boolean RESOURCES_claims
            mandatory
                String RESOURCES_CLAIMS_name
    Boolean CONTAINER_securityContext
    Boolean CONTAINER_startupProbe
        mandatory
            Boolean STARTUPPROBE_exec
                mandatory
                    String STARTUPPROBE_EXEC_command
            Integer STARTUPPROBE_failureThreshold
            Boolean STARTUPPROBE_grpc
                mandatory
                    Integer STARTUPPROBE_GRPC_port
                    String STARTUPPROBE_GRPC_service
            Boolean STARTUPPROBE_httpGet
                optional
                    String STARTUPPROBE_HTTPGET_host
                    Boolean STARTUPPROBE_HTTPGET_httpHeaders
                        mandatory
                            String STARTUPPROBE_HTTPHEADERS_name
                            String STARTUPPROBE_HTTPHEADERS_value
                    String STARTUPPROBE_HTTPGET_path
                    Integer STARTUPPROBE_HTTPGET_port
                    String STARTUPPROBE_HTTPGET_scheme
            Integer STARTUPPROBE_initialDelaySeconds
            Integer STARTUPPROBE_periodSeconds
            Integer STARTUPPROBE_successThreshold
            Boolean STARTUPPROBE_tcpSocket
                mandatory
                    Boolean STARTUPPROBE_TCPSOCKET_host
                    Boolean RSTARTUPPROBE_TCPSOCKET_port
            Integer STARTUPPROBE_terminationGracePeriodSeconds
            Integer STARTUPPROBE_timeoutSeconds
    Boolean CONTAINER_stdin
    Boolean CONTAINER_stdinOnce
    String CONTAINER_terminationMessagePath
    String CONTAINER_terminationMessagePolicy
    Boolean CONTAINER_tty
    Boolean CONTAINER_volumeDevices cardinality [0..*]
        mandatory
            String VOLUMEDEVICES_devicePath
            String VOLUMEDEVICES_name
```

```
        Boolean CONTAINER_volumeMounts cardinality [0..*]
            mandatory
                String VOLUMEMOUNTS_mountPath
                String VOLUMEMOUNTS_name
        Boolean CONTAINER_workingDir
Boolean PODSPEC_dnsPolicy
    alternative
        Boolean DNSPOLICY_ClusterFirstWithHostNet
        Boolean DNSPOLICY_ClusterFirst
        Boolean DNSPOLICY_Default
        Boolean DNSPOLICY_None
Boolean PODSPEC_enableServiceLinks
Boolean PODSPEC_hostIPC
Boolean PODSPEC_hostNetwork
Boolean PODSPEC_hostPID
Boolean PODSPEC_hostUsers
String PODSPEC_hostname
String PODSPEC_nodeName
Boolean PODSPEC_Volumes cardinality [0..*]
    mandatory
        String VOLUMES_name
        Boolean VOLUMES_type {abstract}
            or
                Boolean emptyDir
                Boolean hostPath
                Boolean persistentVolumeClaim
                Boolean configMap
                Boolean secret
                Boolean nfs
                Boolean awsElasticBlockStore
                Boolean gcePersistentDisk
                Boolean azureDisk
                Boolean csi
                Boolean azureFile
                Boolean cephfs
                Boolean cinder
                Boolean downwardAPI
                Boolean ephemeral
                Boolean fc
                Boolean flexVolume
                Boolean flocker
                Boolean glusterfs
```

```
                        Boolean iscsi
                        Boolean photonPersistentDisk
                        Boolean portworxVolume
                        Boolean projected
                        Boolean quobyte
                        Boolean rbd
                        Boolean scaleIO
                        Boolean storageos
                        Boolean vsphereVolume
Boolean ServiceSpec {abstract}
    optional
        Boolean SERVICESPEC_allocateLoadBalancerNodePorts
        Boolean SERVICESPEC_clusterIP {abstract}
            alternative
                Boolean CLUSTERIP_none
                Boolean CLUSTERIP_emptyString
                String CLUSTERIP_IP
        Boolean SERVICESPEC_clusterIPs {abstract}
            alternative
                Boolean CLUSTERIPS_none
                Boolean CLUSTERIPS_emptyString
                String CLUSTERIPS_IPs
        String SERVICESPEC_externalIPs
        String SERVICESPEC_externalName
        Boolean SERVICESPEC_externalTrafficPolicy
            alternative
                Boolean EXTERNALTRAFFICPOLICY_Local
                Boolean EXTERNALTRAFFICPOLICY_Cluster
        Integer SERVICESPEC_healthCheckNodePort
        Boolean SERVICESPEC_internalTrafficPolicy
            alternative
                Boolean INTERNALTRAFFICPOLICY_Local
                Boolean INTERNALTRAFFICPOLICY_Cluster
        Boolean SERVICESPEC_ipFamilies
            optional
                Boolean IPFAMILIES_IPv4
                Boolean IPFAMILIES_IPv6
        Boolean SERVICESPEC_ipFamilyPolicy
            alternative
                Boolean SingleStack
                Boolean PreferDualStack
                Boolean RequireDualStack
```

```
                        String SERVICESPEC_loadBalancerClass
                        Boolean SERVICESPEC_Ports cardinality [0..*]
                            mandatory
                                Integer SERVICESPEC_PORTS_port
                            optional
                                String SERVICESPEC_PORTS_name
                                Boolean SERVICESPEC_PORTS_protocol {abstract}
                                    alternative
                                        Boolean PORTS_PROTOCOL_tcp
                                        Boolean PORTS_PROTOCOL_udp
                                        Boolean PORTS_PROTOCOL_sctp
                                Integer SERVICESPEC_PORTS_nodePort
                                String SERVICESPEC_PORTS_targetPort
                        Boolean publishNotReadyAddresses
                        Boolean SPEC_Selector cardinality [0..*]
                            mandatory
                                String SELECTOR_key
                                String SELECTOR_value
                        Boolean SERVICESPEC_sessionAffinity
                            alternative
                                Boolean SESSIONAFFINITY_ClientIP
                                Boolean SESSIONAFFINITY_None
                        Boolean SERVICESPEC_sessionAffinityConfig
                            mandatory
                                Boolean SESSIONAFFINITYCONFIG_clientIP
                                    mandatory
                                        Integer CLIENTIP_timeoutSeconds
                        String SERVICESPEC_trafficDistribution
                        Boolean SERVICESPEC_Type
                            alternative
                                Boolean TYPE_ExternalName
                                Boolean TYPE_ClusterIP
                                Boolean TYPE_NodePort
                                Boolean TYPE_LoadBalancer
                Boolean ConfigMapSpec {abstract}
                Boolean SecretSpec {abstract}
constraints
    // Restrincciones de Group
    (GROUP_apiextensions | GROUP_apiregistration | GROUP_apps | GROUP_authorization | GROUP_batch |
GROUP_coordination | GROUP_core | GROUP_discovery | GROUP_events | GROUP_node | GROUP_policy |
GROUP_rbac_authorization | GROUP_scheduling) => VERSION_v1
    (GROUP_admission_registration | GROUP_authentication) => VERSION_v1 | VERSION_v1beta1 | VERSION_v1alpha1
```

```
    (GROUP_certificates | GROUP_internal_apiserver | GROUP_networking | GROUP_storage) => VERSION_v1 |
VERSION_v1alpha1

    GROUP_autoscaling => VERSION_v2 | VERSION_v1

    GROUP_flowcontrol_apiserver => VERSION_v1beta3 | VERSION_v1

    GROUP_resource => VERSION_v1alpha2 | VERSION_v1

    // Restricciones de Kind

    (KIND_DaemonSet | KIND_Deployment | KIND_ReplicaSet | KIND_StatefullSet | KIND_ControllerRevision) =>
GROUP_apps

    (KIND_Container | KIND_Pod | KIND_ReplicationController | KIND_Service | KIND_Endpoints | KIND_ConfigMap |
KIND_Secret | KIND_PersistentVolumeClaim | KIND_Volume | KIND_LimitRange | KIND_PodTemplate | KIND_Binding |
KIND_ComponentStatus | KIND_Namespace | KIND_Node | KIND_PersistentVolume | KIND_ResourceQuota |
KIND_ServiceAccount) => GROUP_core

    (KIND_CronJob | KIND_Job) => GROUP_batch

    KIND_EndpintSlice => GROUP_discovery

    (KIND_Ingress | KIND_IngressClass | KIND_IPAddress | KIND_ServiceCIDR | KIND_NetworkPolicy) =>
GROUP_networking

    (KIND_CSIDriver | KIND_CSINode | KIND_StorageClass | KIND_CSIStorageCapacity | KIND_VolumeAttachment |
KIND_VolumeAttributesClass) => GROUP_storage

    (KIND_ClusterTrustBundle | KIND_CertificateSigningRequest) => GROUP_certificates

    KIND_CustomerResourceDefinition => GROUP_apiextensions

    KIND_Event => GROUP_events

    KIND_HorizontalPodAutoscaler => GROUP_autoscaling

    (KIND_MutatingWebhookConfiguration | KIND_ValidatingWebhookConfiguration | KIND_ValidatingAdmissionPolicy
| KIND_ValidatingAdmissionPolicyBinding) => GROUP_admission_registration

    (KIND_PodSchedulingContext | KIND_ResourceClaim | KIND_ResourceClaimTemplate | KIND_ResourceClass |
KIND_ResourceClaimParameters | KIND_ResourceSlice) => GROUP_resource

    KIND_PodDisruptionBudget => GROUP_policy

    KIND_PriorityClass => GROUP_scheduling

    KIND_APIService => GROUP_apiregistration

    (KIND_LocalSubjectAccessReview | KIND_SelfSubjectAccessReview | KIND_SelfSubjectRulesReview |
KIND_SubjectAccessReview) => GROUP_authorization

    (KIND_FlowSchema | KIND_PriorityLevelConfiguration) => GROUP_flowcontrol_apiserver

    KIND_Lease => GROUP_coordination

    KIND_RuntimeClass => GROUP_node

    (KIND_SelfSubjectReview | KIND_TokenRequest | KIND_TokenReview) => GROUP_authentication

    KIND_StorageVersion => GROUP_internal_apiserver

    (KIND_ClusterRole | KIND_ClusterRoleBinding | KIND_Role | KIND_RoleBinding)=> GROUP_rbac_authorization

    // Resto de restricciones

    (TYPE_NodePort | TYPE_LoadBalancer) => (SERVICESPEC_PORTS_nodePort > 0)

    KIND_Pod => PodSpec

    KIND_Deployment => DeploymentSpec & PodSpec

    KIND_Service => ServiceSpec

    KIND_ConfigMap => ConfigMapSpec

    KIND_Secret => SecretSpec

    (maxUnavailable == 0) => (maxSurge > 0)

    DeploymentSpec => PodSpec
```

```
KIND_Service => ServiceSpec

SERVICESPEC_clusterIP => !SERVICESPEC_clusterIPs

SERVICESPEC_clusterIPs => !SERVICESPEC_clusterIP

SERVICESPEC_clusterIP => !TYPE_ExternalName

(SERVICESPEC_externalName == '') => TYPE_ExternalName

SERVICESPEC_ipFamilyPolicy => !TYPE_ExternalName

SERVICESPEC_ipFamilies => TYPE_ClusterIP | TYPE_NodePort | TYPE_LoadBalancer

(SERVICESPEC_healthCheckNodePort > 0) => TYPE_LoadBalancer & EXTERNALTRAFFICPOLICY_Local

(SERVICESPEC_loadBalancerClass == '') => TYPE_LoadBalancer

SESSIONAFFINITY_ClientIP => (CLIENTIP_timeoutSeconds > 0) & (CLIENTIP_timeoutSeconds < 86401)

PODSPEC_activeDeadlineSeconds > 0

LIVENESSPROBE_GRPC_port > 0 & LIVENESSPROBE_GRPC_port < 65535

LIVENESSPROBE_HTTPGET_port > 0 & LIVENESSPROBE_HTTPGET_port < 65535

LIVENESSPROBE_failureThreshold > 0

LIVENESSPROBE_periodSeconds > 0

LIVENESSPROBE_successThreshold > 0

LIVENESSPROBE_terminationGracePeriodSeconds > 0

LIVENESSPROBE_timeoutSeconds > 0

CONTAINER_PORTS_hostPort > 0 & CONTAINER_PORTS_hostPort < 65535

READINESSPROBE_GRPC_port > 0 & READINESSPROBE_GRPC_port < 65535

READINESSPROBE_HTTPGET_port > 0 & READINESSPROBE_HTTPGET_port < 65535

READINESSPROBE_failureThreshold > 0

READINESSPROBE_periodSeconds > 0

READINESSPROBE_successThreshold > 0

READINESSPROBE_terminationGracePeriodSeconds > 0

READINESSPROBE_timeoutSeconds > 0

STARTUPPROBE_GRPC_port > 0 & STARTUPPROBE_GRPC_port < 65535

STARTUPPROBE_HTTPGET_port > 0 & STARTUPPROBE_HTTPGET_port < 65535

STARTUPPROBE_failureThreshold > 0

STARTUPPROBE_periodSeconds > 0

STARTUPPROBE_successThreshold > 0

STARTUPPROBE_terminationGracePeriodSeconds > 0

STARTUPPROBE_timeoutSeconds > 0
```

Anexo B

Comandos para ejecutar los scripts

El script de generación de ficheros para Kubernetes acepta las siguientes opciones:

- --config ó -c: Ruta al archivo de configuración del FM (obligatorio).
- --map ó -m: Ruta al archivo del mapeo (obligatorio).
- --template ó -t: Ruta al archivo plantilla (obligatorio).
- --kubernetes ó -k: Opción de bandera para indicar si se debe comprobar la validez del resultado para Kubernetes (opcional).
- --details: Opción de bandera para dar detalles de optimización del archivo YAML (opcional).
- --dockerfile ó -d: Opción de bandera para indicar si se debe comprobar la validez del resultado para Dockerfile (opcional).

Un ejemplo de la ejecución de este script es el siguiente:

```
python main_resolve_variability.py -c config.yaml -m mapping.yaml -t template.yaml
```