

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

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

por

Enrique López Encinas

24 de junio de 2024

# 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, 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. 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

[Resumen 2](#_Toc170320708)

[1. Introducción 4](#_Toc170320709)

[1.1 Kubernetes 4](#_Toc170320710)

[1.2 Líneas de productos software. 6](#_Toc170320711)

[1.3 Entorno de trabajo y herramientas usadas 7](#_Toc170320712)

[2. Metodología 8](#_Toc170320713)

[2.1 Generación de los ficheros de configuración 8](#_Toc170320714)

[2.2 Resultados del análisis de variabilidad del FM de Kubernetes 8](#_Toc170320715)

[3. Conclusiones 10](#_Toc170320716)

[Referencias 11](#_Toc170320717)

[Modelo de características de Kubernetes 12](#_Toc170320718)

[Comandos para ejecutar los scripts 27](#_Toc170320719)

# 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:

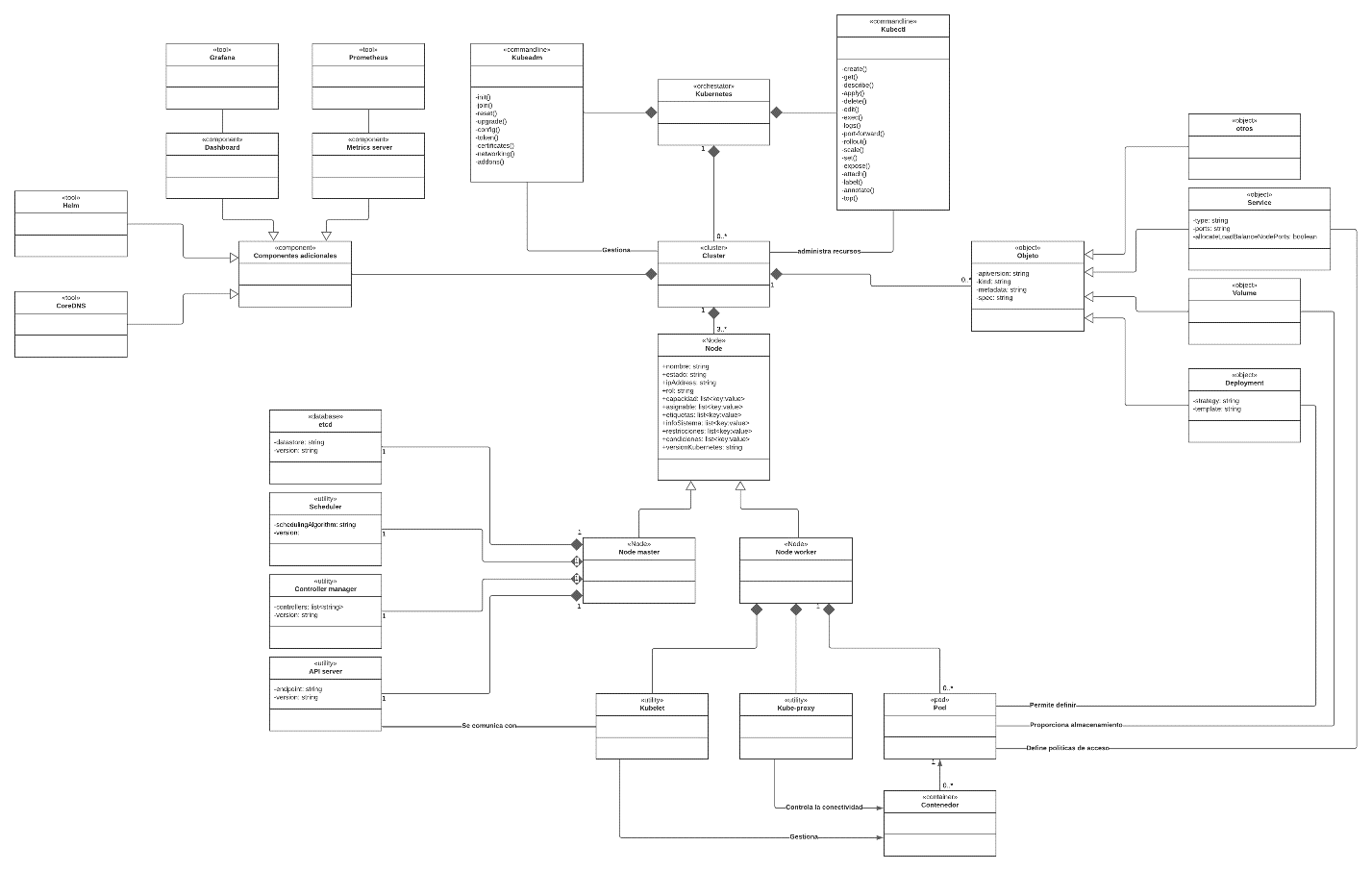
## Kubernetes

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.

## 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.

Diagrama

Descripción generada automáticamente

**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 configuraciónválida 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]](#footnote-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.

# 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 artículo, se ha desarrollado una primera versión del FM de Kubernetes, que 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 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 Junja2 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.

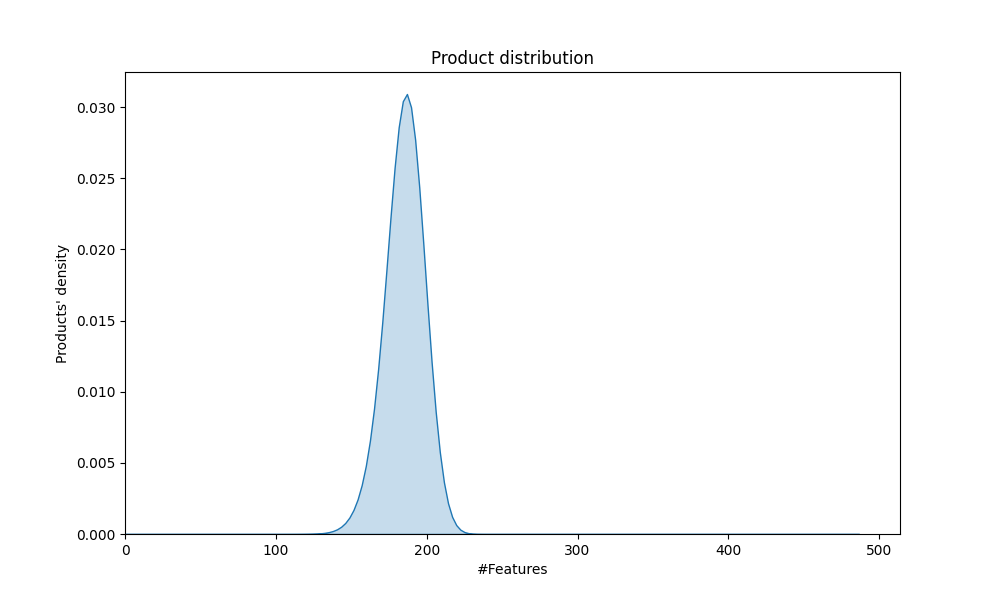
Una vez se disponen de estos ficheros, con ayuda de un script en Python basado en el código del GitHub uvengine (Jmhorcas. (s. f.)) y cuyo código, con ligeras modificaciones, se puede encontrar en GitHub fms\_dataset (CAOSD-group. (s. f.).) se obtiene el fichero de configuración listo para desplegarlo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

Como paso adicional, se pueden usar herramientas de validación como Kubeconform (Yannh. (s. f.)) o Kube-score [[2]](#footnote-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.

## 2.2 Resultados del análisis de variabilidad del FM de Kubernetes

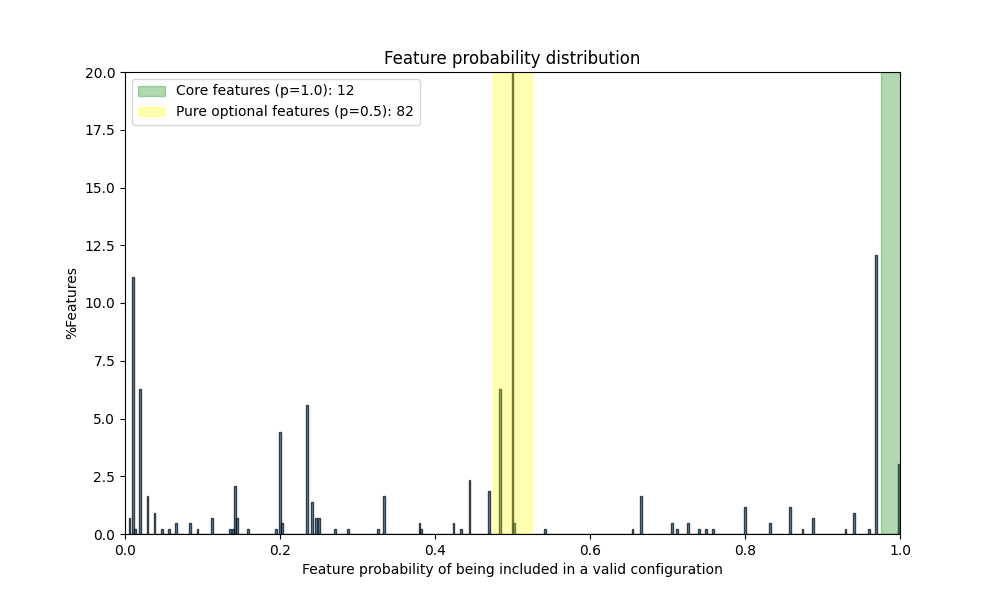
Usando la herramienta Flamapy ya mencionada anteriormente, se ha conseguido analizar la variabilidad del FM de Kubernetes, obteniendo los siguientes datos en esta primera versión:

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 mas a la derecha se encuentre, mas probable es que esa característica se incluya en una configuración valida. Por ejemplo, vemos a la izquierda un numero 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 numero de configuraciones posibles a 6.78e42 configuraciones validas aproximadamente.

# 3. 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*

1. Página oficial de Flamapy: https://flamapy.github.io/ [↑](#footnote-ref-1)
2. Herramienta online de Kube-score: https://kube-score.com/ [↑](#footnote-ref-2)