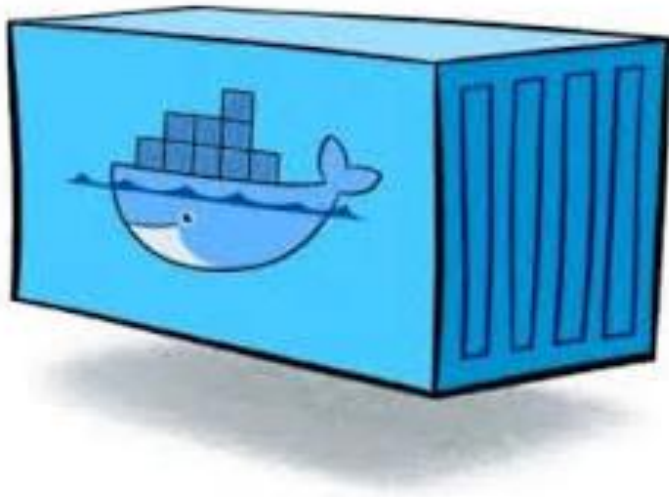


2022

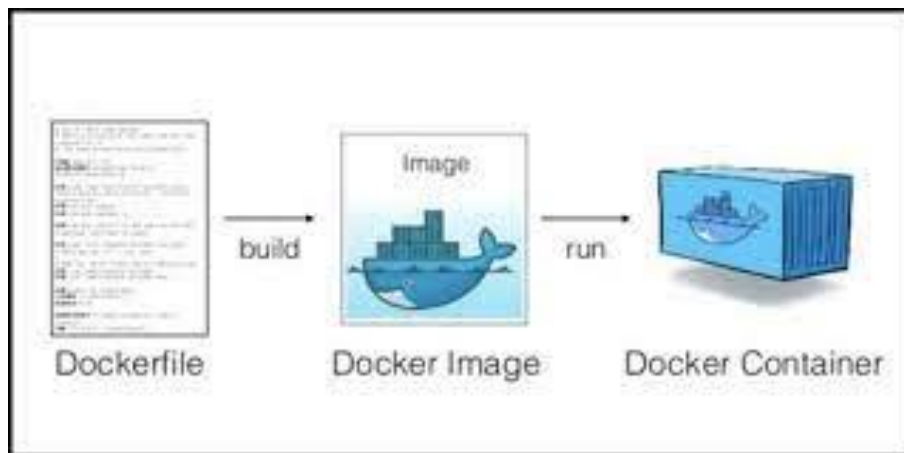
KUI (Kubernetes User Interface)



Abraham Zambrano Ferrer
Administración de
Sistemas Informáticos en
Red

13-12-2022

Palabras Clave

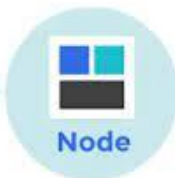


Contenedores

Docker

Kubernetes

Python



kubernetes



docker

Índice

Palabras Clave.....	2
Índice.....	3
Definición del proyecto	4
Conceptos Básicos	4
Contenedores	4
Kubernetes.....	6
Docker	6
Podman.....	6
Pods.....	6
Biblioteca	7
Python.....	7
Git.....	7
GitHub.....	7
WSL.....	8
Visual Studio Code.....	8
Explicación Bibliotecas	8
Instalación Bibliotecas.....	9
Textualize.....	9
Rich.....	9
Plot Text	10
Dive	10
Poetry.....	10
Instalación del entorno	11
Go	11
Docker	11
Kind	12
Instalación de un Contenedor	12
Manual de Usuario	13
Puesta en escena KUI	14
Trabajo Futuro	17
Conclusiones.....	17
Bibliografía.....	18
Contraportada	21

Definición del proyecto

El proyecto KUI es una aplicación de interfaz de línea de comandos (CLI, *Command Line Interface*), diseñada para facilitar la visualización de eventos en sistemas de orquestación de contenedores Kubernetes.

KUI (Kubernetes User Interface) muestra un menú interactivo a través del cual el operador del clúster puede navegar a través de los pods de los distintos namespaces y consultar los logs de las distintas aplicaciones mientras se están ejecutando.

KUI cuenta con una [licencia X11 \(también conocida como MIT\)](#) es decir: permite re-utilizar el software dentro de otro software propietario. Haciendo posible que todo el mundo pueda conocer esta aplicación, usarla, mejorarla e implementarla en otras aplicaciones.

El desarrollo de KUI se realiza en la plataforma GitHub y el repositorio es [totalmente público](#):



Conceptos Básicos

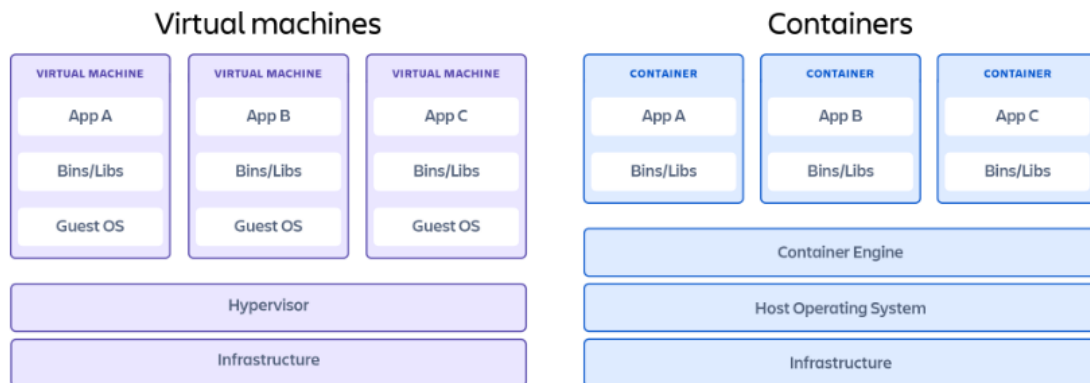
Como prólogo de las diferentes funcionalidades del proyecto, comienzo a explicar varios conceptos esenciales para conocer en mayor profundidad las diferentes funciones de la aplicación.

Contenedores

Los contenedores son un método de aislamiento de recursos. De esta forma, se empaqueta una aplicación y todas sus dependencias (bibliotecas de las que depende, otros binarios, ficheros de configuración, variables de entorno...) y se pone a ejecutar en cualquier lugar donde exista un engine de contenedores.

La virtualización es ese proceso por el cual, sistemas software simulan las funcionalidades de sistemas físicos (memoria, CPU...). Esto se hace para hacer un mayor aprovechamiento de los recursos físicos y mediante el aislamiento de recursos entre aplicaciones.

Diferencias entre máquinas virtuales y contenedores



Antes de la llegada de los contenedores, las máquinas virtuales se utilizaban para poner sistemas en producción. De esta forma, cuando se precisaba de hacer un escalado horizontal de la aplicación, un orquestador de máquinas virtuales (como Vagrant) creaba nuevas copias de la máquina virtual que ejecutaba la aplicación. Una vez la carga se reducía, el orquestador de máquinas virtuales escalaba hacia abajo eliminando las máquinas virtuales sobrantes.

Actualmente, los contenedores se utilizan con el mismo propósito, utilizando orquestadores de contenedores como Kubernetes, OpenShift o Apache Mesos.

Mientras las máquinas virtuales precisan de un hipervisor de tipo 1 (donde se instala el hipervisor directamente sobre el hardware) o de tipo 2 (donde se instala sobre un sistema operativo), los contenedores utilizan mecanismos de aislamiento de recursos propios de Linux (como cgroups). Esto conlleva a un mayor aprovechamiento de recursos (pudiendo lanzar más instancias de nuestras aplicaciones en los recursos antes utilizados por el hipervisor).

Algunas de las ventajas del uso de frente al de máquinas virtuales a la hora de poner aplicaciones en producción son:

- **Menos sobrecarga**
Los contenedores requieren menos recursos software y hardware que las máquinas virtuales. Esto nos permite poder lanzar más instancias de la misma aplicación en una misma máquina.
- **Mayor portabilidad**
Las aplicaciones que se ejecutan en contenedores se pueden poner en marcha fácilmente en sistemas operativos y plataformas de hardware diferentes.

- **Mayor eficiencia**
Los contenedores permiten poner en marcha, aplicar parches o escalar las aplicaciones con mayor rapidez.
- **Mejor desarrollo de aplicaciones**
Los contenedores respaldan los esfuerzos ágiles y de DevOps para acelerar los ciclos de desarrollo, prueba y producción.

Kubernetes

Kubernetes es un orquestador de contenedores. Permite poner aplicaciones containerizadas en producción, garantizando alta disponibilidad y facilitando tareas como rollbacks, migración entre nubes, escalado horizontal y otras muchas features. Kubernetes es open source y desarrollada por empresas como Google, Red Hat, Microsoft, Amazon o Alibaba. Muchos servicios cloud ofrecen la posibilidad de desplegar clusters Kubernetes en su infraestructura o utilizar clusters totalmente gestionados por el proveedor de nube (como GKE en Google Cloud o AKS en Azure).

Docker

Docker es un engine para contenedores, es decir, un motor para contenedores creado por la empresa Docker.inc. Es uno de los más utilizados en el mundo, aunque está cayendo en desuso en favor de Podman.

Mientras que los contenedores están pensados para sistemas operativos Linux, la versión de escritorio de Docker tiene soporte para sistemas operativos Mac (creando contenedores Linux utilizando un hipervisor de tipo 2) y Windows (creando contenedores Windows y Linux a través de Hyper-V).

Podman

Podman es otro container engine, totalmente Open Source y desarrollado por Red Hat, para desarrollar, gestionar y ejecutar los contenedores en los sistemas Linux.

Podman gestiona el ecosistema completo de contenedores usando la biblioteca libpod. Y su funcionamiento se encuentra en alza al no necesitar permisos de superusuario para lanzar nuevos contenedores ni el uso de un Daemon.

Pods

Los pods son los objetos más pequeños y básicos que se pueden implementar en Kubernetes.

Un pod representa una instancia única de un proceso en ejecución en un clúster. Los pods contienen uno o más contenedores.

Biblioteca

Una biblioteca es un paquete software que permite a los desarrolladores reutilizar lógica entre sus distintos desarrollos. Los desarrollos pueden descargar y utilizar bibliotecas implementadas por otros desarrolladores (o empresas) sin preocuparse del funcionamiento interno de dichos módulos ni de cómo tienen que ser testeados.

Python

Python es un lenguaje de programación de alto nivel, orientado a objetos. Es ampliamente utilizado para implementar automatización de sistemas, así como para implementar lógica de negocio. Es un lenguaje en auge ya que es muy usado para desarrollar aplicaciones orientadas a machine learning y data science.

Python es relativamente simple, ya que requiere una sintaxis única que se centra en la legibilidad, por tanto, esto reduce el costo de mantenimiento y de desarrollo de programas. Permite que los equipos trabajen en colaboración y sin barreras significativas de lenguaje y experimentación.

Además, soporta el uso de módulos y paquetes, lo que significa que los programas pueden ser diseñados en un único estilo y el código puede ser reutilizado en varios proyectos.

Git

Git es un sistema distribuido de control de versiones, gratuito y de código abierto. Fue concebido con la idea de ofrecer un gran rendimiento, ocupar menos espacio en disco y evitar la necesidad de un servidor central para que pueda ser distribuido. Actualmente es el sistema de control de versiones más utilizado del mundo.

GitHub

GitHub es una de las principales plataformas para crear proyectos abiertos de herramientas y aplicaciones, y se caracteriza sobre todo por sus funciones colaborativas que ayudan a que todos puedan aportar su granito de arena para mejorar código.

Por lo tanto, es una plataforma la cual permite subir, apoyar, compartir y descargar proyectos. A parte sirve como punto de referencia a la reutilización del código que propone la programación orientada a objetos.

WSL

WSL o también conocido como Windows Subsystem for Linux, es una característica que nos permite a los usuarios de Windows usar el bash de la distribución de Linux que elijas, podemos tener los directorios típicos de Linux como son "Home" o "Bin" en Windows y los directorios de Windows en el bash de Linux, incluye la mayoría de herramientas de línea de comando, utilidades y aplicaciones, sin modificar y sin la sobrecarga de una máquina virtual tradicional o una configuración de arranque dual.

Visual Studio Code

VS Code es un editor de código fuente desarrollado por Microsoft, es software libre y multiplataforma, además cuenta con una muy buena integración con Git, cuenta con soporte para depuración de código y un sinnúmero de extensiones.

Este editor incluye una terminal con todas las funciones, la cual se inicia fácilmente en el directorio de trabajo. La terminal integrada puede utilizar cualquier Shell instalado en el equipo.

Explicación Bibliotecas

Textualize

Textual es un framework de Python para crear aplicaciones interactivas que se ejecutan en un terminal.

Rich

Rich es una biblioteca de Python, específica para Textual que proporciona un texto enriquecido y un formato bonito para la terminal.

Plotext

Plotext es una biblioteca de Python que permite realizar gráficas interactivas para la terminal.

Dive

Dive es una herramienta para explorar los Dockers, revisar todos sus ficheros con sus diferentes permisos y descubrir formas de reducir el tamaño de estas imágenes.

Poetry

Poetry es un empacador de dependencias para Python, este empacador nos ayuda a declarar, administrar e instalar dependencias/bibliotecas de proyectos de Python, lo que garantiza que tenga la pila correcta en todas partes.

Instalación Bibliotecas

Textualize

Instalación Textualize

```
pip install "textual[dev]" -> Esta opción instala unas pequeñas dependencias para desarrolladores
```

```
pip install textual
```

Podemos también clonar el repositorio para utilizar ese código para nuestros proyectos.

Para clonarlo utilizaremos git (A continuación, se explica detalladamente -> Plot Text)

Rich

Al instalar textual, también se te instala esta librería, pero su instalación sería la siguiente:

```
python -m pip install rich
```

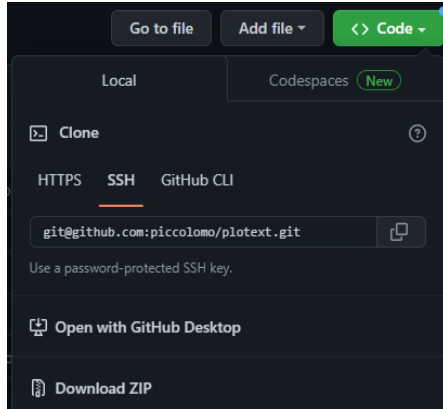
Podemos también clonar el repositorio para utilizar ese código para nuestros proyectos.

Para clonarlo utilizaremos git (A continuación, se explica detalladamente -> Plot Text)

Plot Text

Esta instalación es un poco más compleja, pero habría que realizar estos pasos:

<https://github.com/piccolomo/plotext> -> ir al botón verde donde pone code y copiar el ssh.



Una vez eso entraremos a GIT, nos ponemos en la ruta donde queremos que se coloque esta carpeta y pondremos el siguiente comando:

```
git clone "el ssh copiando anteriormente"
```

Ya tendremos el repositorio clonado y podremos comenzar a usarlo.

Dive

Para tener esta biblioteca, lo único que tendríamos que hacer es clonar el repositorio como he mostrado previamente.

Poetry

Poetry es una dependencia de Python, la cual te permite declarar las bibliotecas de tu proyecto y te la instala automáticamente.

El proceso de instalación sería el siguiente:

Instalar poetry:

```
curl -sSL https://install.python-poetry.org | python3 -
```

Añadirlo a tu PATH:

```
$HOME/.local/bin
```

Instalación del entorno

Go

Para instalar go solo tendremos que ir a la página oficial (<https://go.dev/>) y descargar la última versión.

Docker

Para la instalación de docker deberemos hacer lo siguiente:

```
sudo apt update  
sudo apt upgrade
```

Instalar el paquete de requisitos

```
sudo apt-get install curl apt-transport-https ca-certificates software-properties-common
```

Agregar los repositorios

Descarga

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo  
apt-key add -
```

Agregamos

```
sudo add-apt-repository "deb [arch=amd64] https://download.do-  
cker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Actualizamos

```
sudo apt update
```

Instalamos

```
sudo apt install docker-ce
```

Kind

KiND (Kubernetes in Node Docker) es una herramienta que nos permite lanzar un entorno Kubernetes para desarrollo. Lo utilizaremos en este proyecto de forma que podamos mostrar la utilidad de KUI de forma práctica.

Instalación de un Contenedor

Para la creación de un contenedor predefinido, he utilizado la herramienta KIND (explicada anteriormente), Kubectl y Docker.

Para la creación y administración, estos son los comando correspondientes:

Creación de un clúster (Con el nombre predefinido que seria “kind”)

```
kind create cluster
```

Creación de un clúster con un nombre diferente

```
kind create cluster - --name “el-nombre-que-quieras”
```

Comprobar clústers/contenedores creados

```
kind get clusters
```

```
docker ps
```

Comprobar los nodos

```
kind get nodes
```

Comprobar los namespaces

```
kubectl get namespaces
```

Comprobar los pods de un determinado namespaces

```
kubectl get pods -n “el-namespaces-que-quieras”
```

Comprar los logs de un pod determinado

```
kubectl logs “el-pod-que-quieras” -n “el-namespaces-de-ese-pod”
```

Manual de Usuario

Para instalar esta herramienta sería conveniente seguir los siguientes pasos:

1º Actualización de la lista de paquetes disponibles y sus versiones:

`"sudo apt update"`

2º Instalación las nuevas versiones respetando la configuración del software:

`"sudo apt upgrade"`

3º Instalación del paquete "git":

`"sudo apt-get install git"`

4º Instalación de las diferentes bibliotecas (Instalación de bibliotecas)

5º Descarga una copia de KuiApp a tu dispositivo:

Por Https:

`"git clone https://github.com/AbrahamZambrano/KUI.git"`

Por SSH:

`"git clone git@github.com:AbrahamZambrano/KUI.git"`

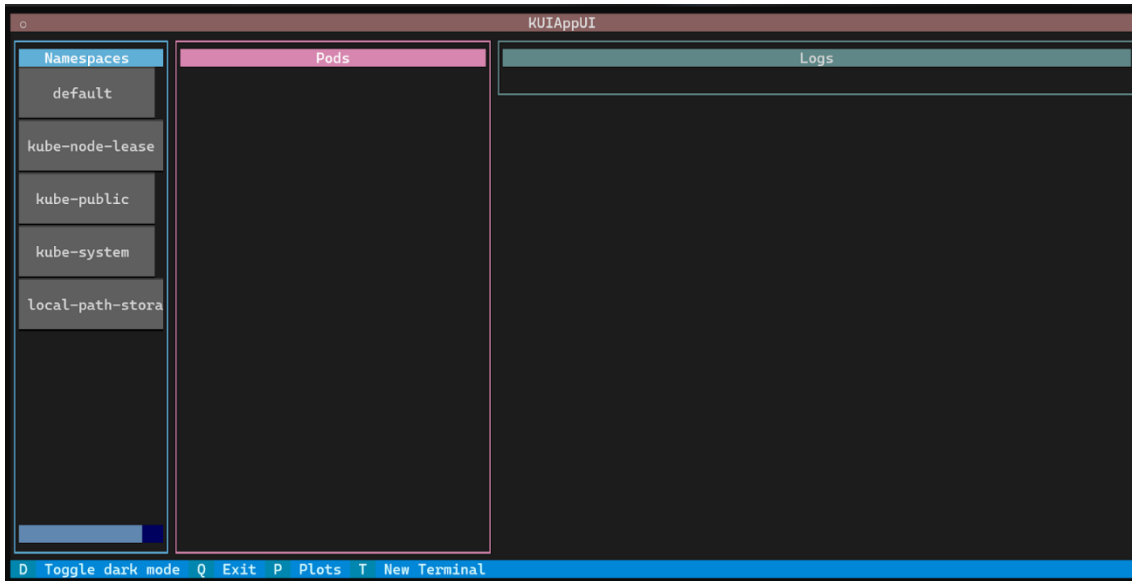
6º Cambiar el directorio al de dentro del paquete.

`"cd KuiApp"`

7º Ejecutar el siguiente comando para el comienzo de la aplicación:

`"poetry run task kuiapp"`

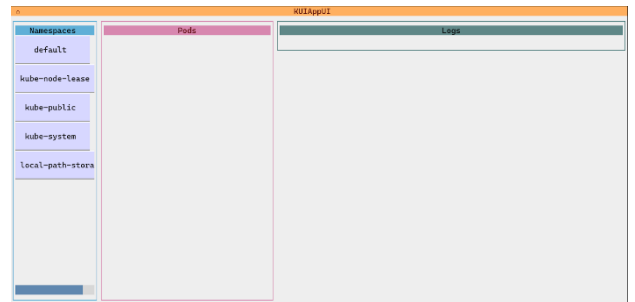
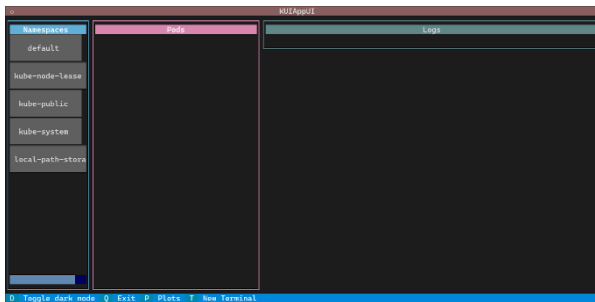
Puesta en escena KUI



Este sería el menú principal de la aplicación el cual está distribuido en 3 partes y un footer:

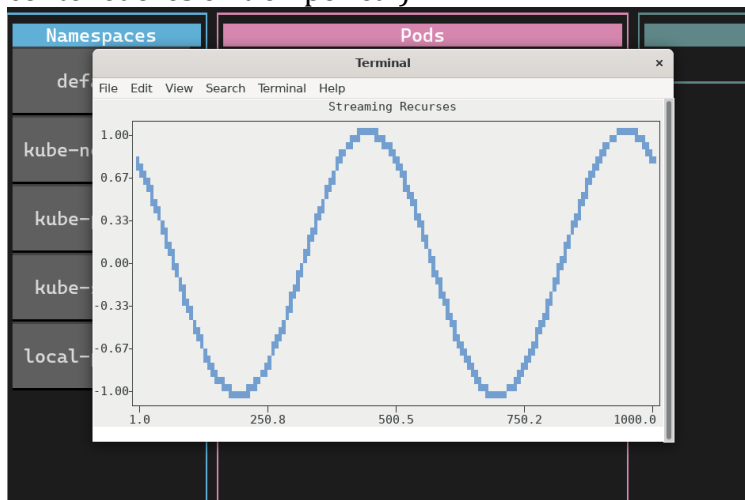
- La primera parte, consta de un espacio reservado solo para los namespaces o los clusters de tipo virtual que son respaldados por un mismo clúster físico.
- La segunda parte, consta de los pods (Conjunto de varios contenedores) del namespace que selecciones.
- La tercera parte, consta de los ficheros de logs del pod que se haya seleccionado
- El footer que consta de varias acciones:

“D” -> Activar o desactivar el modo oscuro

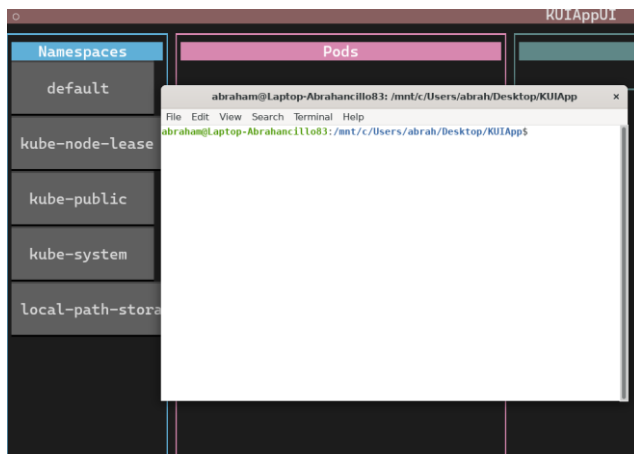


“Q” -> Salir de la Aplicación

“P” -> Plots (Nos enseña una gráfica con los recursos que están gastando los contenedores en tiempo real)



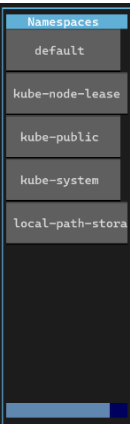
“T” -> Nos abre una nueva Terminal en la ruta en la que nos encontremos



¡IMPORTANTE!

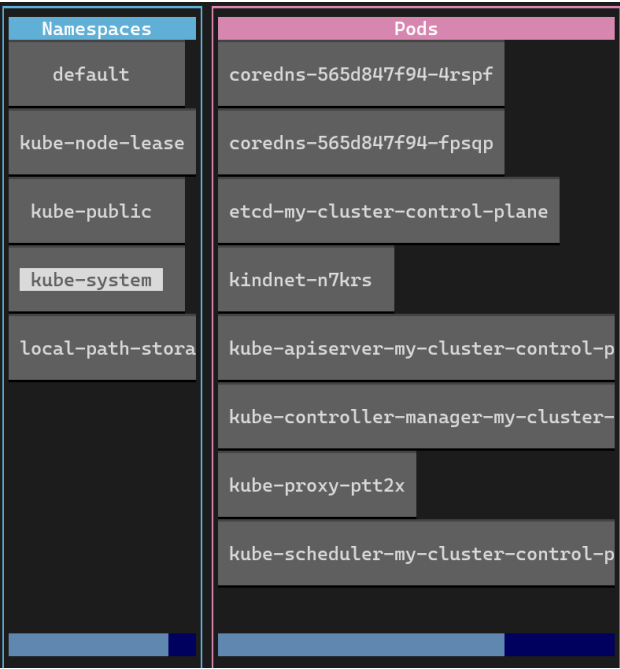
Tener los contenedores inicializados, ya que Kui, está conectada directamente con los contenedores, si estos no están inicializados, la aplicación aparecerá vacía.

Una vez ejecutada la aplicación, nos saldrá directamente los namespaces de todos los kubernetes que dispongamos.



Namespaces
default
kube-node-lease
kube-public
kube-system
local-path-storage

Si seleccionamos cualquiera de los namespaces, automáticamente nos saldrán todos los Pods de este en concreto:



Namespaces	Pods
default	coredns-565d847f94-4rspf
kube-node-lease	coredns-565d847f94-fpsqp
kube-public	etcd-my-cluster-control-plane
kube-system	kindnet-n7krs
local-path-storage	kube-apiserver-my-cluster-control-p
	kube-controller-manager-my-cluster-
	kube-proxy-ptt2x
	kube-scheduler-my-cluster-control-p

Si seleccionamos cualquiera de estos pods nos saldrá su archivo de logs en concreto:

Namespaces		Pods		Logs	
default		coredns-565d847f94-4rsfp		I1205 17:22:35.805056	1 main.go:316] probe TCP address
kube-node-lease		coredns-565d847f94-fpsqp		my-cluster-control-plane:6443	
kube-public		etcd-my-cluster-control-plane		I1205 17:22:35.807433	1 main.go:102] connected to apiserver:
kube-system		kindnet-n7krs		https://my-cluster-control-plane:6443	
local-path-storage		kube-apiserver-my-cluster-control-p		I1205 17:22:35.807461	1 main.go:107] hostIP = 172.18.0.2
		kube-controller-manager-my-cluster-		podIP = 172.18.0.2	
		kube-proxy-ptt2x		I1205 17:22:35.807557	1 main.go:116] setting mtu 1500 for CNI
		kube-scheduler-my-cluster-control-p		I1205 17:22:35.807579	1 main.go:146] kindnetd IP family: "ipv4"
				I1205 17:22:35.807613	1 main.go:150] noMask IPv4 subnets:
				[10.244.0.0/16]	
				I1205 17:22:36.100933	1 main.go:223] Handling node with IPs:
				map[172.18.0.2:{}]	
				I1205 17:22:36.100968	1 main.go:227] handling current node
				I1205 17:22:46.114507	1 main.go:223] Handling node with IPs:
				map[172.18.0.2:{}]	
				I1205 17:22:46.114551	1 main.go:227] handling current node
				I1205 17:22:56.123686	1 main.go:223] Handling node with IPs:
				map[172.18.0.2:{}]	
				I1205 17:22:56.123721	1 main.go:227] handling current node
				I1205 17:23:06.133179	1 main.go:223] Handling node with IPs:
				map[172.18.0.2:{}]	
				I1205 17:23:06.133206	1 main.go:227] handling current node
				I1205 17:23:16.144656	1 main.go:223] Handling node with IPs:
				map[172.18.0.2:{}]	
				I1205 17:23:16.144682	1 main.go:227] handling current node
				I1205 17:23:26.149410	1 main.go:223] Handling node with IPs:
				map[172.18.0.2:{}]	

Trabajo Futuro

- Integración con la feature para debugging de pods corriendo en Kubernetes: <https://kubernetes.io/docs/tasks/debug/debug-application/debug-running-pod/>
- Asegurar su funcionamiento con OpenShift: <https://www.redhat.com/en/technologies/cloud-computing/openshift>
- Mejorar el formateo del código añadiendo el uso de linternas
- Añadir Continuous Integration al repositorio de la herramienta
- Añadir nuevos plots para obtener nuevas estadísticas
- Integrar Dive para mejorar la depuración de las imágenes: <https://github.com/wagoodman/dive>

Conclusiones

Este proyecto me ha hecho crecer en el mundo de la programación ya que he aprendido a programar en un lenguaje totalmente desconocido para mí, como era Python, aparte de que hace tener un buen perfil laboral debido a su compleja planificación, realización y distribución.

Demuestra el tipo de competencias que tiene dicha persona.

Bibliografía

<https://github.com/piccolomo/plotext>

<https://github.com/wagoodman/dive>

<https://github.com/Textualize/textual>

<https://github.com/Textualize/rich>

<https://code.visualstudio.com/docs/?dv=win64user>

https://textual.textualize.io/getting_started/

<https://learn.microsoft.com/es-es/windows/wsl/tutorials/gui-apps>

<https://askubuntu.com/questions/250929/pythonpath-environment-variable>

<https://textual.textualize.io/help/>

<https://stackoverflow.com/questions/50202238/python-pip-requestsdependencywarning-urllib3-1-9-1-or-chardet-2-3-0-doe>

<https://sleeplessbeastie.eu/2021/10/29/how-to-inspect-and-improve-docker-image-using-dive-utility/>

<https://towardsdatascience.com/json-and-textual-data-in-python-5aa7c376a0d4>

<https://github.com/piccolomo/plotext/blob/master/readme/utilities.md#command-line-tool>

<https://github.com/piccolomo/plotext/blob/master/readme/subplots.md#subplots>

<https://www.textualize.io/textual/gallery>

<https://github.com/kraanzu/gupshup>

[https://github.com/mle-infrastructure/mle-monitor](https://github.com/mle-infrastructure/mle-monitor/blob/main/mle_monitor/dashboard/components/plots.py)

<https://github.com/mle-infrastructure/mle-monitor>

<https://help.wnpower.com/hc/es/articles/360048910771-C%C3%B3mo-instalar-Docker-en-tu-servidor-con-Ubuntu>

<https://ubunlog.com/go-instala-este-lenguaje-de-programacion-ubuntu-20-04/>

<https://kind.sigs.k8s.io/>

https://kubernetes.io/es/docs/tasks/debug-application-cluster/_print/

https://onthedock.github.io/post/170414-error_the-connection-to-the-server-was-refused/

<https://www.netapp.com/es/devops-solutions/what-are-containers/>

https://www.redhat.com/es/topics/containers/learning-kubernetes-tutorial?sc_cid=7013a000002wLvzAAE&gclid=CjwKCAiAvK2bBhB8EiwAZUbP1HD4MPRfGnVjYWzyTEdybx5LPgW2j54UJg- NM 9tvMEoYW-3iqCMBocUa0QAvD BwE&gclsrc=aw.ds

https://www.redhat.com/es/topics/containers/what-is-docker?sc_cid=7013a000002wLvzAAE&gclid=CjwKCAiAvK2bBhB8EiwAZUbP1NiVdvlsgxNeBiMfA2sS1I3FXQqD8y1BxTHo0vf1L9BIILRcU TuBoCDKUQAvD BwE&gclsrc=aw.ds

<https://www.javiergarzas.com/2015/07/que-es-docker-sencillo.html>

<https://immune.institute/blog/librerias-python-que-son/>

<https://www.cursosaula21.com/que-es-python/>

<https://www.atlassian.com/es/git/tutorials/what-is-git>

<https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>

<https://platzi.com/tutoriales/1748-terminal/2349-que-es-y-como-usar-windows-subsystem-for-linux/>

<https://learn.microsoft.com/es-es/windows/wsl/about>

https://github.com/Textualize/textual/blob/main/tests/test_widget_mounting.py

https://textual.textualize.io/styles/display/#_tabbed_1_1

<https://stackoverflow.com/questions/6034662/python-method-overriding-does-signature-matter>

https://rich.readthedocs.io/en/stable/search.html?q=RenderableType&check_keywords=yes&area=default

<https://textual.textualize.io/api/widget/>

<https://github.com/Textualize/textual/blob/main/examples/calculator.css>

<https://github.com/Textualize/textual/blob/main/examples/calculator.py>

<https://regex101.com/>

<https://textual.textualize.io/tutorial/>

<https://stackoverflow.com/questions/2710940/python-if-x-is-not-none-or-if-not-x-is-none>

<https://www.askpython.com/python/built-in-methods/python-frozenset>

<https://github.com/microsoft/WSL/issues/8824>

<https://github.com/microsoft/WSL/issues/7883>

<https://github.com/microsoft/WSL/issues/8170>

<https://www.geeksforgeeks.org/python-program-to-convert-a-list-to-string/>

https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/python-sleep/?ac=OM.WE.WEo42K356300T7073a&itc=L0Q5C23R-FAC1E9-&utm_source=google&utm_medium=cpc&utm_campaign=SGE-ES-MYW-MIXX---PERFORMANCE_MAX---

[&utm_term=&matchtype=&utm_content=&gclid=CjwKCAiA68ebBhB-EiwALVC-NtMcMzkbarV31jvliQ98Y4FRjvzalya7rrO3ulsWku_5oFDTxBMjJxoCzCkQAvD_BwE&gclid=aw.ds](#)

[https://www.atlassian.com/es/microservices/cloud-computing/containers-vs-vms#:~:text=La%20diferencia%20clave%20entre%20los,del%20nivel%20del%20sistema%20operativo](#)

[https://www.google.com/search?q=que+son+los+Pods+contenedores&rlz=1C1FHFK_eS978ES978&sxsrf=ALiCzsYJTvB4pQEuztr2H4257EFSt9Ecog%3A1668504792087&ei=2FzY_OwBlmbkgXbi5yQDQ&ved=0ahUKEwjz_Lb8K_7AhWJjaQKHdsFB9IQ4dUDCA8&uact=5&oq=que+son+los+Pods+contenedores&gs_lcp=Cgxnd3Mtd2l6LXNlcnAQAzIFCAAQogQyBQgAEKIEMgUIABCiBDIFCAAQogQ6CggAEecQ1gQQsAM6BAgiECc6BQgAEIAEOgYIABAWEB46BAghEBU6CAghEByQHhAdSgQITRgBSgQIQRgASgQIRhgAULMFWN4VYPgWaAFwAXgAgAGPAogB-BaSAQYxLjAuMTGYAQCGAQHIAQjAAQE&sclient=gws-wiz-serp](#)

[https://www.redhat.com/es/topics/containers/what-is-podman#los-pods](#)

[https://www.ionos.es/digitalguide/servidores/herramientas/tutorial-de-podman/#:~:text=Podman%20es%20un%20software%20para,de%20forma%20r%C3%A1pida%20y%20sencilla.](#)

[https://pandorafms.com/blog/es/que-es-podman/](#)

[https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/](#)

[https://textual.textualize.io/guide/queries/](#)

[https://textual.textualize.io/guide/widgets/#_tabbed_5_1](#)

[https://github.com/Textualize/textual/pull/1146](#)

[https://github.com/Textualize/textual/releases](#)

[https://github.com/hidetatz/kubecolor](#)

[https://www.w3.org/wiki/CSS/Properties/color/keywords](#)

[https://textual.textualize.io/guide/layout/#_tabbed_7_3](#)

[https://textual.textualize.io/styles/display/](#)

Contraportada

Aquí os proporciono mis datos

Nombre:

Abraham

Apellidos:

Zambrano Ferrer

Correo:

abrahancillo83@gmail.com

Teléfono:

689308762

Ubicación:

Cenes de la vega, Granada

Github:

<https://github.com/AbrahamZambrano/KUI>