Introducción – Interfaz gráfica

El uso de la herramienta kubectl permite realizar todos los trabajos necesarios para administrar los componentes de “**kubernetes**”. Pero, no se puede negar que el uso de una interfaz gráfica ayuda bastante en la comprensión de los componentes creados para “**kubernetes**”. Existen diversas alternativas para permitir la visualización gráfica de los componentes, en este tutorial, veremos algunas de esas alternativas.

[Kubernetes Dashboard (kube-dashboard) 2](#_Toc129179233)

[LENS 6](#_Toc129179234)

[Octant 9](#_Toc129179235)

[Kubenav 13](#_Toc129179236)

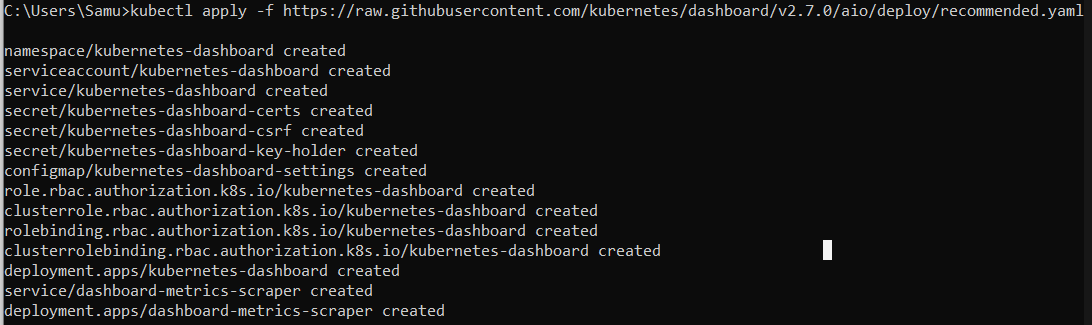
[OpenLens 13](#_Toc129179237)

[Otras aplicaciones 14](#_Toc129179238)

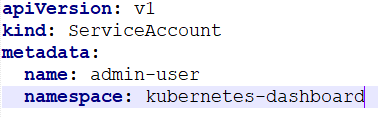
[Consideraciones finales 14](#_Toc129179239)

# Kubernetes Dashboard (kube-dashboard)

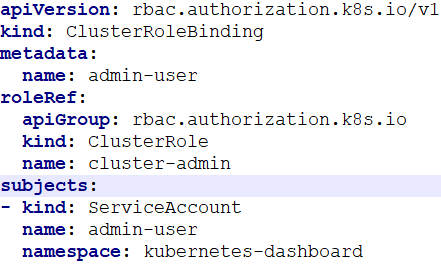
* Instalar:
  + Kubectl:
    - kubectl apply -f <https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml>



* + - Cuando hagamos esto, tenemos que esperar que se descargue la imagen del dashboard.
  + Helm:
    - <https://artifacthub.io/packages/helm/k8s-dashboard/kubernetes-dashboard>.
* Acceder:
  + Primero inicializamos el proxy:
    - kubectl proxy
  + Posteriormente acedemos a la siguiente url:
    - <http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login>
* Crear usuario:
  + La primera vez que accedamos al dashboard, se nos pedirá el uso de un token de autenticación. Para esto, tenemos que crear los siguientes archivos YML.
    - **Crear usuario**:



* + - * Kind: hace referencia el tipo de objeto que estaremos creando.
      * Metadata: datos de información como el nombre o namespace en este caso.
    - **Vincular el rol a un cluster**:



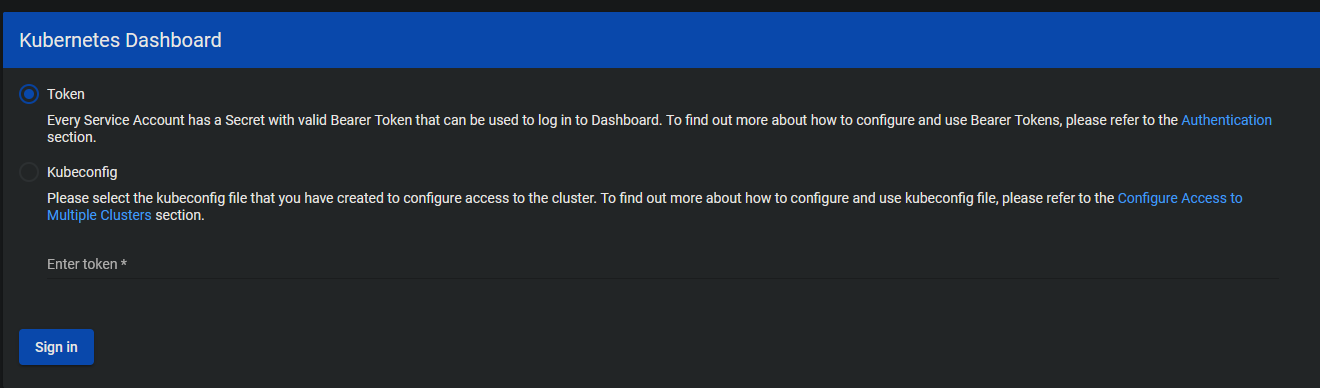
* + - * roleRef: pondremos todos los datos correspondientes al rol que se va a realizar sobre el “**cluster**”
      * Subject: aquí apuntamos al “**service** **account”** creado anteriormente.
    - Ejecutar los dos archivos anteriores:
      * kubectl apply -f ./admin-user.yml



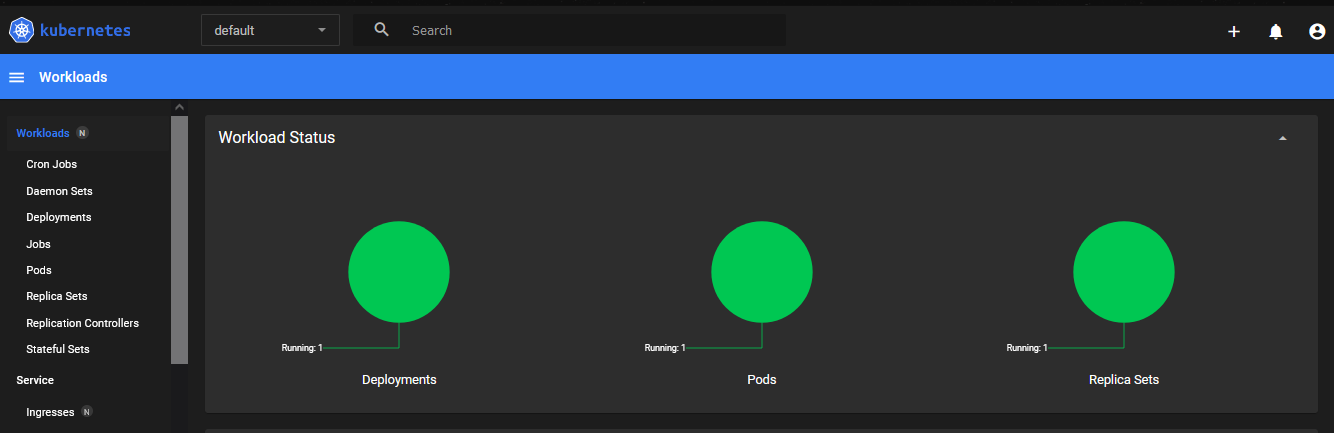
* + - * kubectl apply -f ./rbca.yml



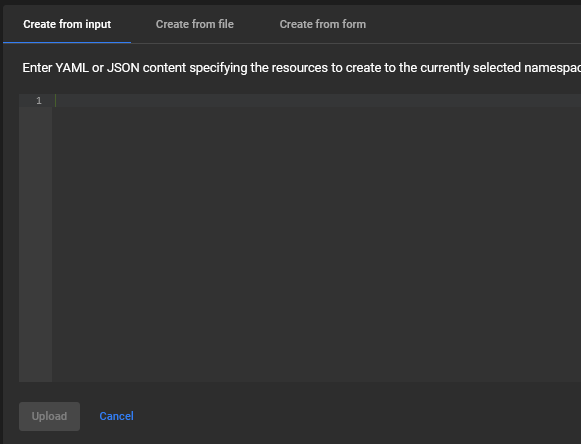
* + - Ahora temenos que generar el token de autenticación para nuestra cuenta.
      * kubectl -n NAMESPACE create token USER
  + Una vez ejecutado el comando de crear token, copiamos el “**output**” o salida del comando y lo pegamos en la interfaz de kube-admin:



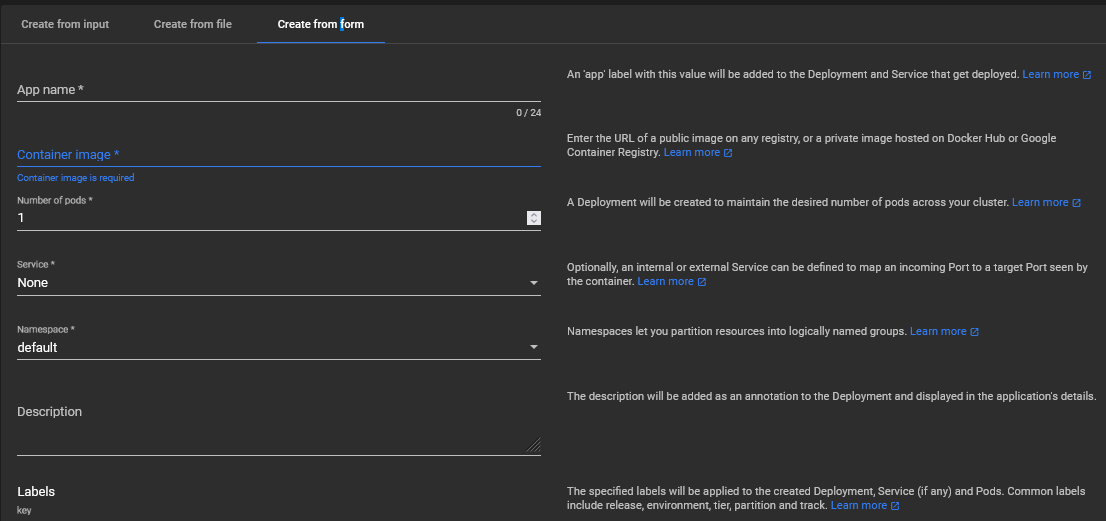
* Una vez dentro, veremos la siguiente interfaz:



* + Este “**dashboard**” está dividido en dos secciones. A la izquierda un menú donde podremos ver todos los objetos dentro de nuestro sistema. A la derecha la información correspondiente al apartado del menú que seleccionemos.
* Además de lo anterior, podemos utilizar el asistente de creación de objetos propios de kube-admin. Para acceder a este presionamos el botón de [**+**] en el menú superior ().
  + Esto nos mostrará la siguiente sección dividida en tres pestañas:



* + - **Create from input**: nos brinda un cuadro de texto donde podremos crear nuestros archivos YML para la creación de “**deployments**”, “**service**” u otros objetos.
    - **Create from file**: nos permite cargar archivos yml ya creados.
    - **Create from form**: en caso que no queramos usar código, es posible llenar todos los datos que necesitemos para crear nuestro “**deployment**” mediante el uso de un formulario:

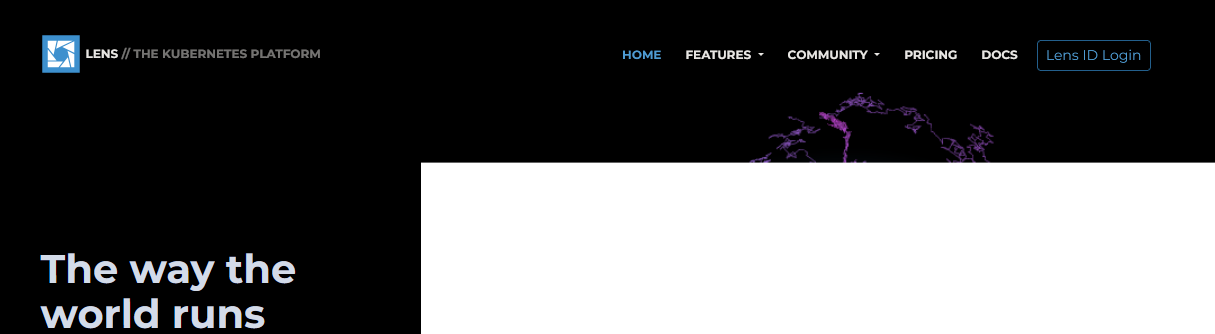


* Como GUI para kubernetes, kube-dashboard es bastante completo y simple de utilizar. Pero seguiremos probando otras alternativas.

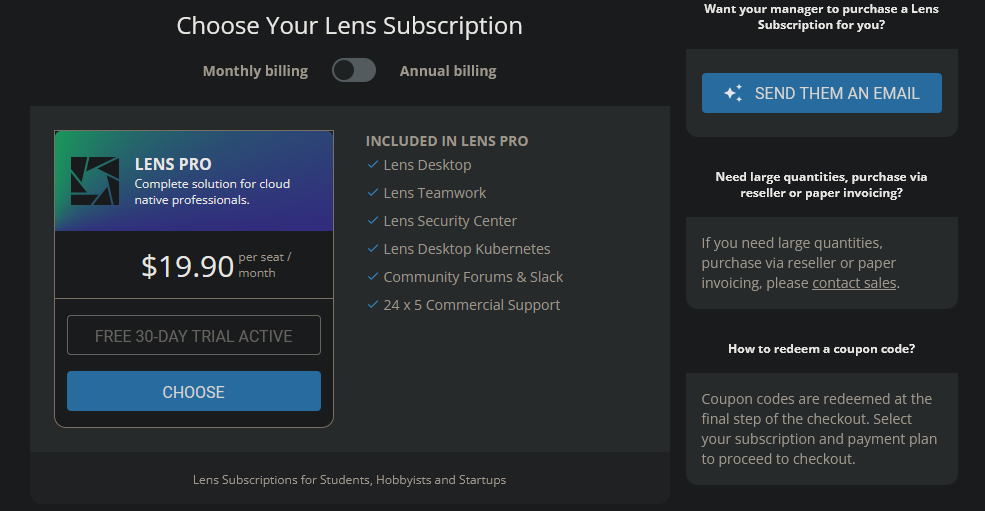
# LENS

Es una aplicación basada en electron (electron posee algunos problemas con el manejo de la memoria) que nos permite administrar mediante una interfaz visual nuestros **cluster** de **kubernetes**. Para instalarlo, tenemos que acceder al sitio web oficial y descargarnos la aplicación:

* <https://k8slens.dev/>



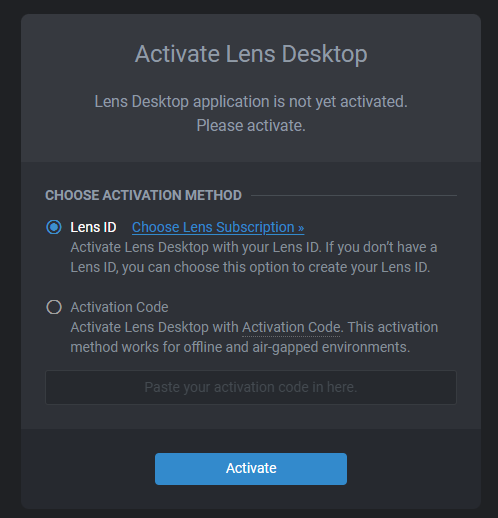
Una vez descargado, instalamos la aplicación como un ejecutable de Windows. Al estar basada en electron, es posible utilizarlo en sistemas Linux y Macos. En Windows específicamente, después de instalada, debemos ir al sitio web y crearnos una cuanta. Para este fin, podemos utilizar nuestra cuenta de Google o de Github. Con la cuenta ya creada, inicializamos nuestra aplicación “**desktop**” y mediante el navegador nos dirigimos a la página <https://app.k8slens.dev/subscribe>.



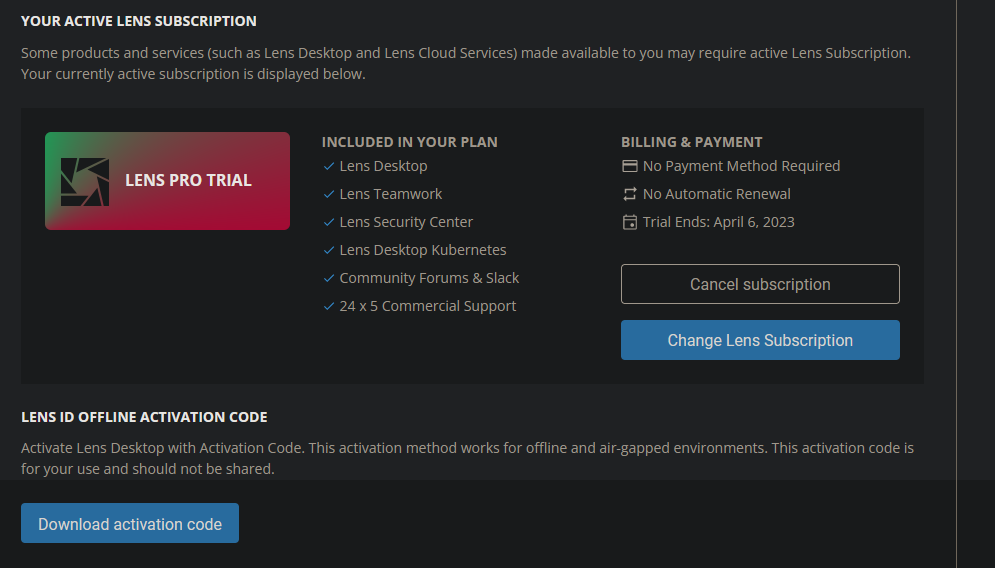
Para usar Lens, es necesario estar subscrito a un tipo de cuenta en específico. Por defecto, nos saldrá la versión pro (para este tutorial, se inicializó el Trial de 30 días) o la versión para estudiantes. A esta última es posible acceder mediante el enlace:



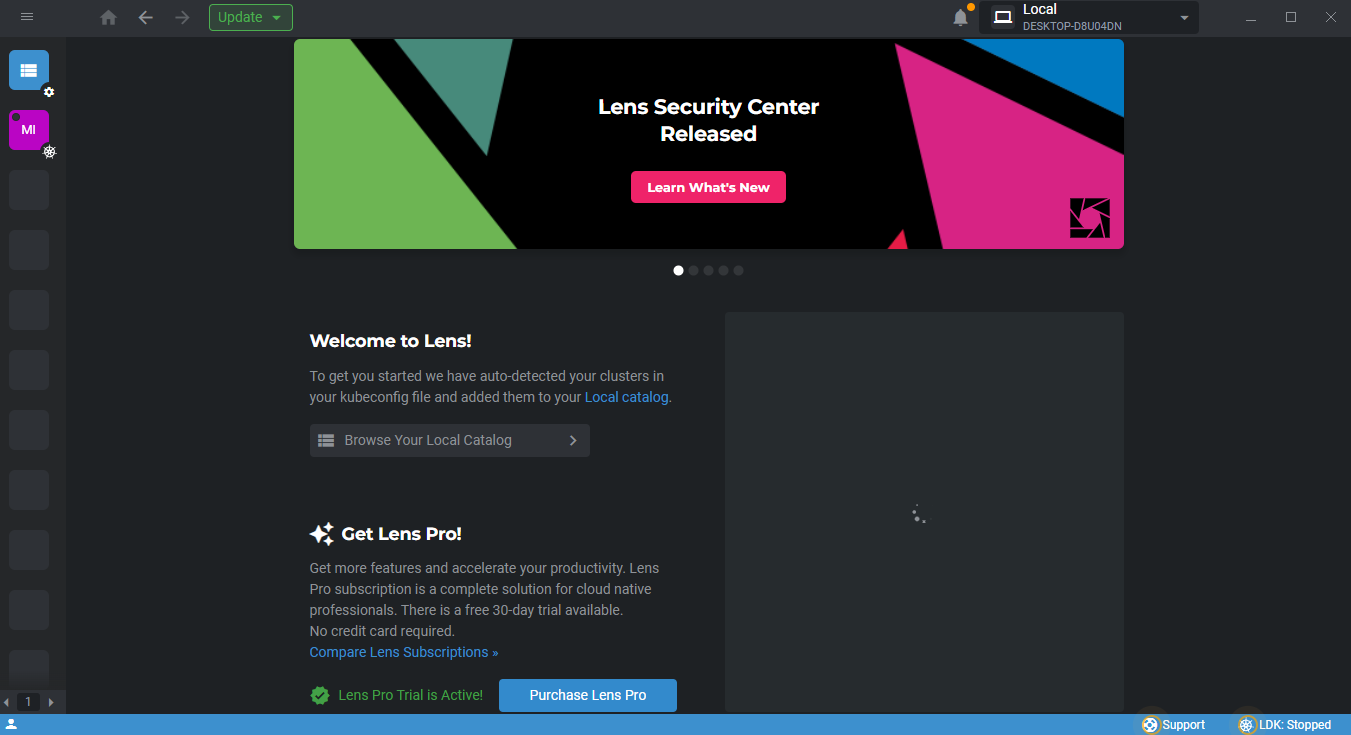
Sin importar que decidamos seleccionar, tendremos que dirigirnos a nuestra aplicación de escritorio y seleccionar entre activación mediante código o Lens ID:



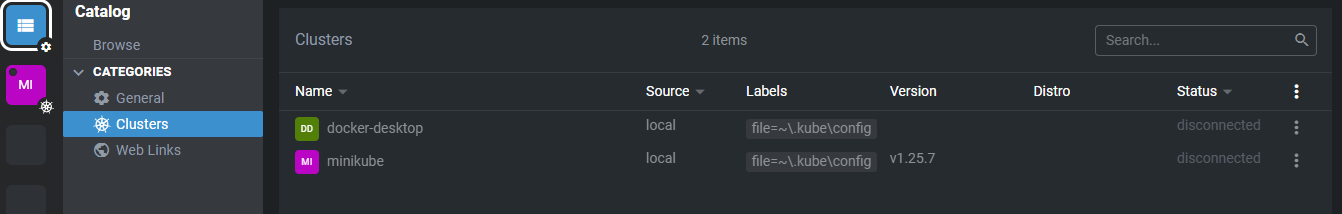
* Si seleccionamos la primera opción, se nos habilitará una pestaña en el navegador que de forma automática comprobará la subscripción que tengamos. En caso de no tener ninguna, nos pedirá que la seleccionemos.
* La activación mediante código solo es posible cuando tengamos ya la subscripción realizada. Nos dirigimos a nuestra página como usuario de Lens y buscamos el botón de “**Download Activation**”:



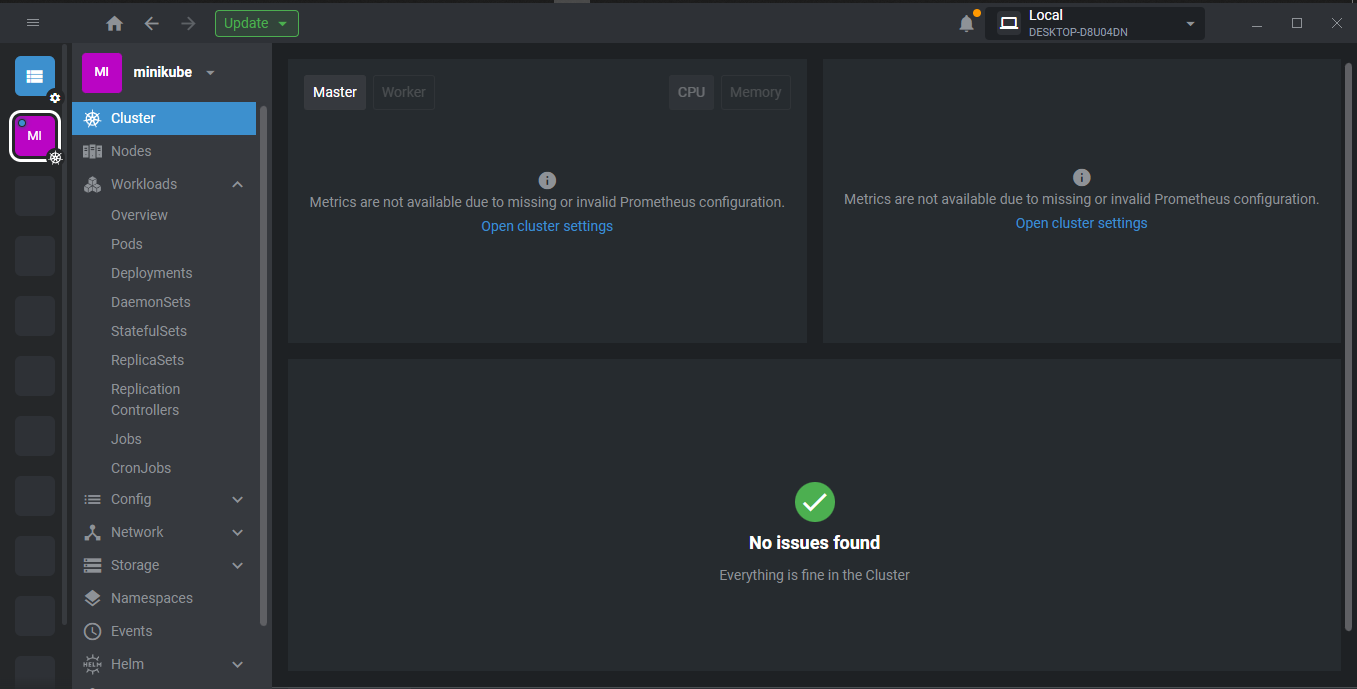
* + Al presionarlo, nos descargará un TXT con nuestro código de activación. Lo copiamos y lo pegamos dentro del recuadro debajo de este tipo activación en la aplicación de escritorio.
* Una vez dentro, la aplicación de Lens tendrá la siguiente forma:



* Para seleccionar el cluster donde vamos a trabajar, presionamos el primer ícono que se nos muestra en la barra lateral izquierda:



* + Dentro buscamos la categoría “**clusters**” y seleccionamos en este caso “**minikube**”.
  + De forma automática, Lens creará un acceso directo en la barra lateral.
* Para acceder a “**minikube**”, debemos presionar sobre su ícono y de esta forma comenzará el proceso de conexión. Una vez dentro, se nos mostrará la siguiente interfaz:



* + En el menú de la izquierda podemos encontrar todos los objetos que podemos manipular dentro de kubernetes y en el centro de la aplicación, toda la información relacionada con la opción seleccionada

# Octant

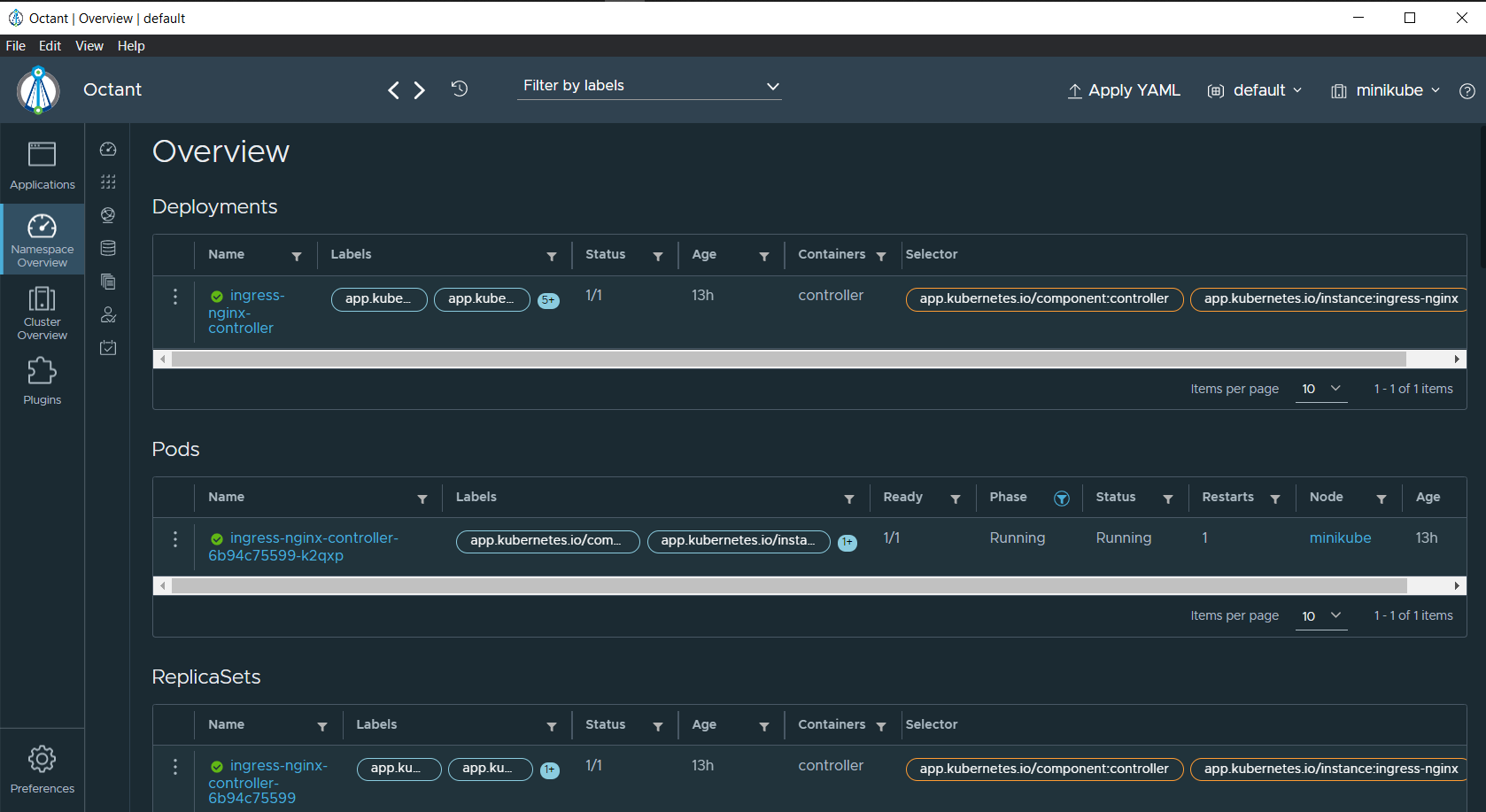
Es una herramienta multiplataforma que nos permite administrar nuestro “**cluster**” de kubernetes. Para descargar esta herramienta, vamos a su sitio web oficial y descargamos el instalador:

* https://octant.dev/

O podemos ir directamente a su repositorio en github y descargar el “**release**” que más se ajuste a nuestras necesidades:

* <https://github.com/vmware-archive/octant/releases/tag/v0.25>.

Una vez descargado, pasamos al proceso de instalación, es bastante sencillo y prácticamente lo único que se tiene que hacer es presionar siguiente en todas las ventanas. Terminada la instalación y abierto el software se nos presentará una interfaz gráfica parecida a la siguiente:



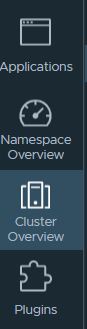
Por defecto, Octan intentará buscar los “**cluster**” disponibles. En este caso tenemos 2, uno ejecutándose sobre “**minikube**” y otro mediante “**docker**”. Se tomará por defecto el “**cluster**” al que “**kubectl**” esté apuntando mediante la variable “**context**”; en nuestro caso “**minikube**”. Si queremos cambiar el contexto, presionamos el botón:



* + Seleccionamos el contexto que queramos y esperamos que el cambio surta efecto.

La interfaz gráfica está dividida en 3 secciones:

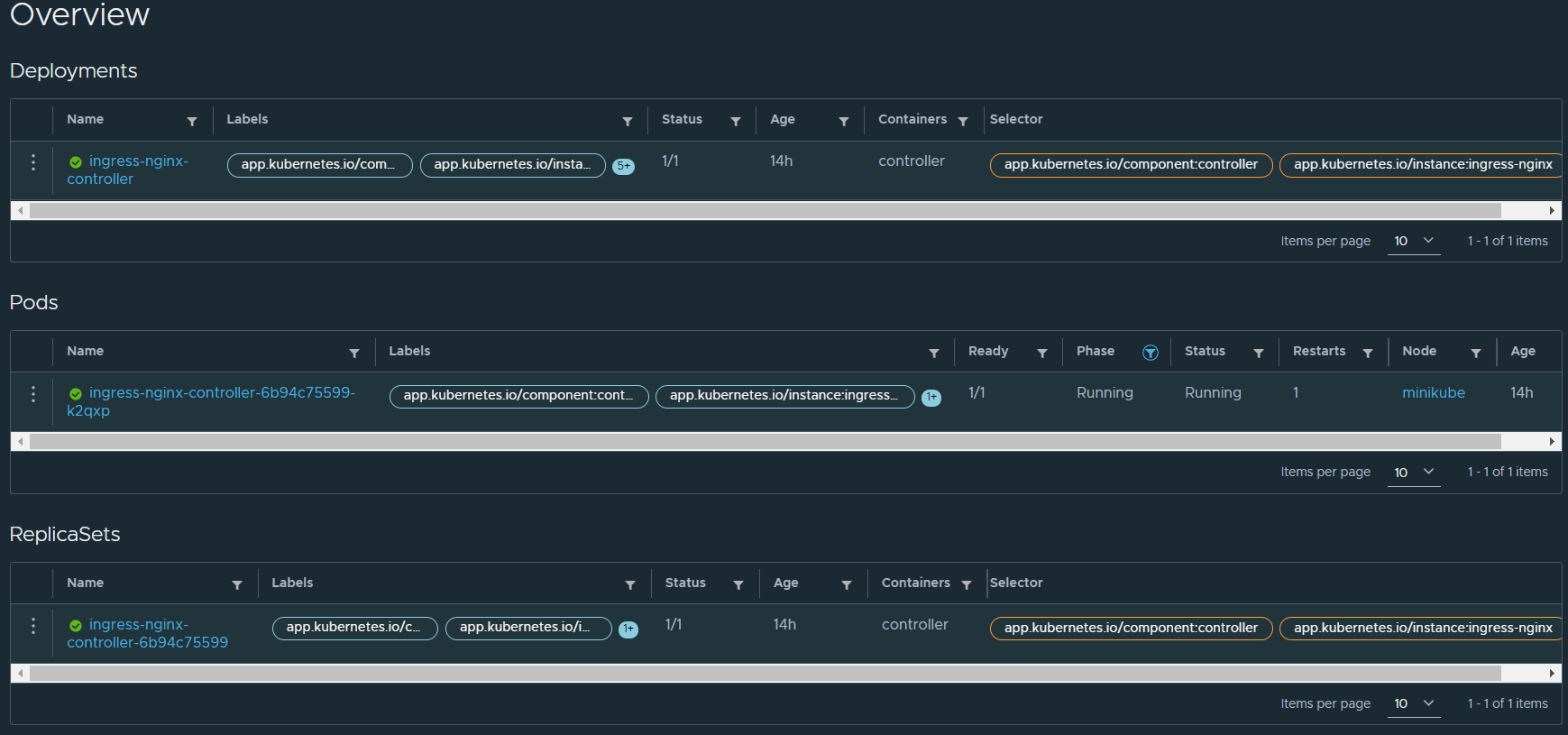
* Barra lateral izquierda: aquí se encuentras las agrupaciones por categoría superior de kubernetes (“**cluster**” y “**namespace**”).



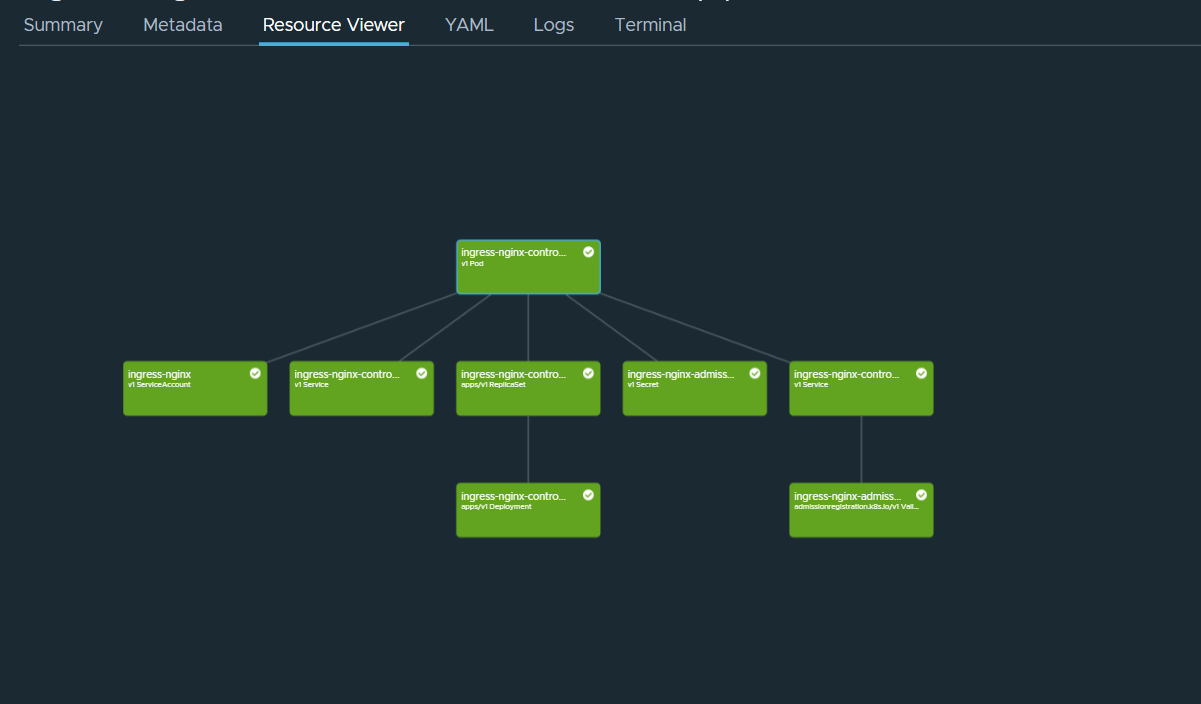
* Barra lateral izquierda 2: Según la opción que seleccionemos del punto anterior, los valores de este menú cambiarán. Aquí tendremos acceso a opciones como “**pods**”, “**deployments**”, etc.



* En el centro: Veremos toda la información correspondiente a la opción seleccionada en la barra lateral izquierda 2.



Octane es una alternativa bastante completa como GUI para kubernetes. Entre sus principales ventajas para el trabajo, es la capacidad de modificar los YML de los objetos que tengamos creados y a su vez, actualizar los pods. Una funcionalidad curiosa de esta herramienta, es la capacidad de ver nuestros objetos en forma de grafo:



* + Este en específico es la representación gráfica del pod del “**ingress controller**” de “**nginx**”

# Kubenav

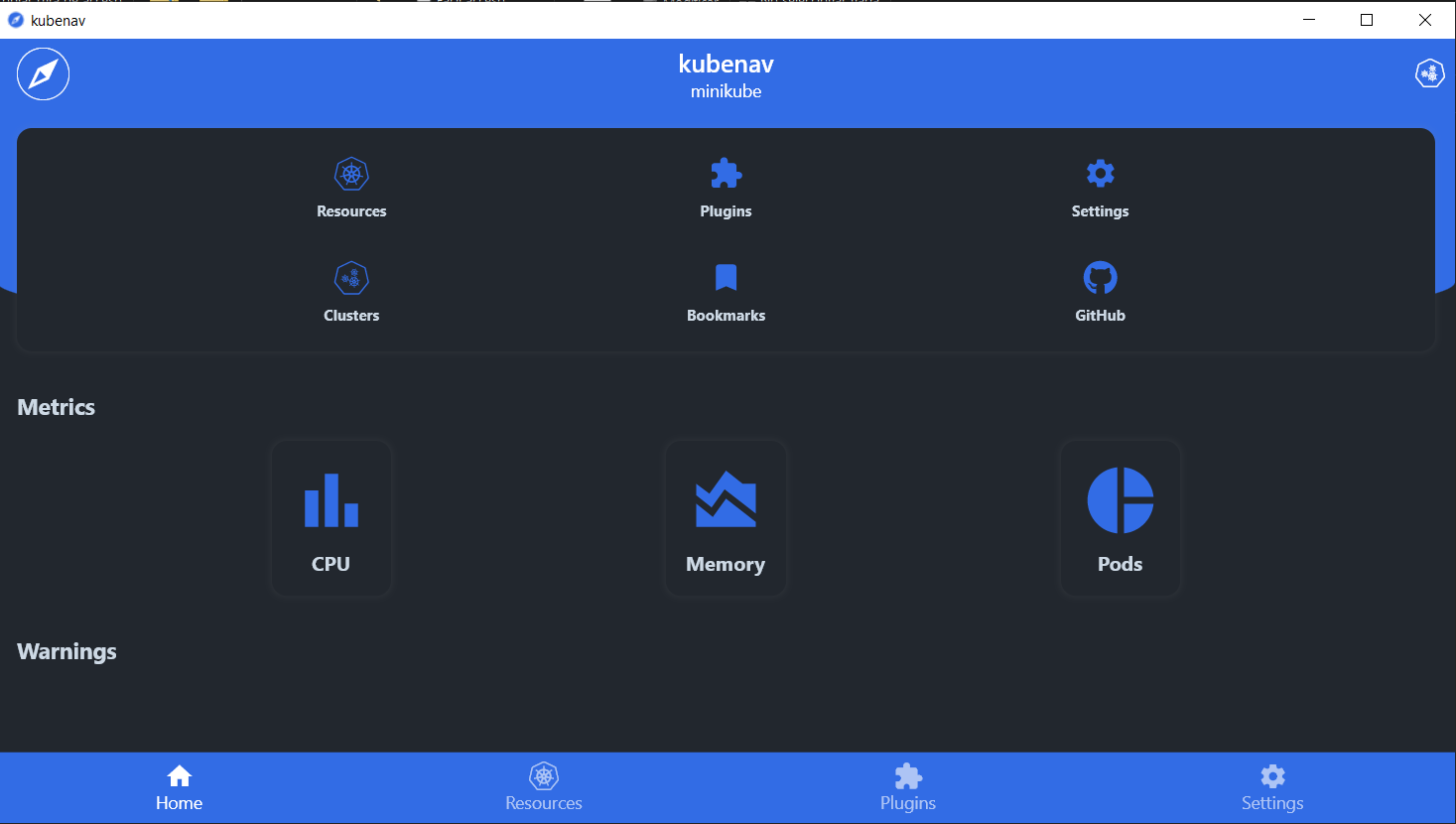
Este es una alternativa ligera para la administración de los “**cluster**” de kubernetse. Entre sus principales ventajas está la capacidad de ofrecer una aplicación móvil para la administración de los “**clusters**”. Para descargar la aplicación nos podemos dirigir al sitio web:

* <https://kubenav.io/>

Aquí nos permitirá seleccionar el dispositivo en el cuál queremos instalar la aplicación. En nuestro caso seleccionamos la opción de “**desktop**” y esto nos lleva a la siguiente página:

* <https://github.com/kubenav/kubenav/releases>

Descargamos la aplicación que vamos a utilizar; en caso de Windows, no hace falta instalarla, la ejecutamos y nos aparecerá la siguiente interfaz de usuario:



En comparación con las dos anteriores, la interfaz gráfica es bastante básica, pero permite realizar operaciones similares como el escalado o modificación del archivo YML.

# OpenLens

Anteriormente vimos la aplicación de LENS, el problema con esta era que estaba sujeta a licencia. OpenLens es una alternativa libre a esta aplicación. Para descargarla vamos al siguiente sitio web:

* <https://github.com/MuhammedKalkan/OpenLens/releases>

Las funcionalidades son bastante similares. La principal diferencia radica en que no es necesario la creación de una cuenta para utilizarla.

# Otras aplicaciones

* Weave scope
  + <https://github.com/weaveworks/scope/releases>
* Konstellate
  + <https://github.com/jeremykross/konstellate?ref=thechiefio>
* Rancher
  + <https://www.rancher.com/>

# Consideraciones finales

En mi opinión creo que las dos mejores alternativas como interfaz gráfica son Lube-dashboard y Lens o OpenLens. La primera principalmente por brindar la capacidad de crear “**deployments**” desde un formulario. La segunda por brindar una interfaz gráfica ordenada y limpia para el trabajo002E