

Pasos:

Primero hice los archivos de configuración YAML para desplegar y gestionar una base de datos MySQL en un clúster de Kubernetes.

Para ello hice y ejecuté el **db-secret.sh** para que genere el **db-secret.yaml**. Este archivo crea un objeto Secret en Kubernetes llamado **db-secret**. Un objeto Secret es utilizado para almacenar información sensible. En este caso, el Secret almacena las credenciales de la base de datos MySQL en forma de datos codificados en base64.

Para desplegar el recurso

```
ubuntu@reyes-nube ~$ kubectl apply -f db-secret.yaml
```

```
ubuntu@reyes-nube:~$ kubectl get secret
NAME          TYPE      DATA      AGE
db-secret      Opaque    3          5d15h
metabase-secret Opaque    4          4d18h
```

Después hice el **db-sts.yaml**. Este archivo crea un objeto StatefulSet llamado **db-sts**, que es una forma de desplegar aplicaciones con persistencia en un clúster de Kubernetes. En este caso, se está utilizando para desplegar y gestionar la base de datos MySQL.

El **db-secret** proporciona las credenciales seguras necesarias para que los Pods dentro del **db-sts** puedan acceder y autenticarse correctamente en la base de datos MySQL que se está ejecutando. La relación entre ambos se establece a través de las variables de entorno en el archivo de configuración del StatefulSet.

Para desplegar el recurso

```
ubuntu@reyes-nube ~$ kubectl apply -f db-sts.yaml
```

```
ubuntu@reyes-nube:~$ kubectl get sts,pod,pvc
NAME          READY  AGE
statefulset.apps/db-sts  1/1    3d19h

NAME          READY  STATUS   RESTARTS  AGE
pod/db-sts-0   1/1    Running  0         3d19h
pod/kb-prod-7c589b995b-5nqmp  1/1    Running  0         5d13h
pod/metabase-76dd58b4c5-gdd22  1/1    Running  0         3d19h

NAME          STATUS  VOLUME          CAPACITY  ACCESS MODES  STORAGECLASS  AGE
persistentvolumeclaim/mysql-data  Bound  pvc-b2a0bc56-1248-47ee-87c9-5348f09841ac  1Gi  RWO  longhorn  56d
persistentvolumeclaim/mysql-data-db-sts-0  Bound  pvc-61374c6c-8a64-40e8-b475-7d793b975188  10Gi  RWO  longhorn  3d19h
persistentvolumeclaim/mysql-volume-db-sts-0  Bound  pvc-4a4c7b2f-3763-4702-b37c-367a1d0869da  5Gi  RWO  longhorn  3d19h
```

Ahora copio el **google-mobility.sql** de mi VM al interior de la carpeta **/docker-entrypoint-initdb.d** del pod **db-sts-0** para que cada vez que se inicie dicho pod ejecute el **.sql**. Esto persiste en **mysql-volume-db-sts-0**.

```
ubuntu@reyes-nube ~$ kubectl cp /home/ubuntu/caso3/google-mobility.sql db-sts-0:/docker-entrypoint-initdb.d/google-mobility.sql -c wp-sitio-db -n martingreyes-dev
```

Y por último hice el **db-svc.yaml**. Este archivo crea un objeto Service llamado **db-svc**, que permite exponer la base de datos MySQL dentro del clúster o en el exterior. Es decir, el **db-svc** proporciona una manera de acceder a los Pods individuales del **db-sts** que ejecutan la base de datos MySQL.

Para desplegar el recurso

```
ubuntu@reyes-nube ~$ kubectl apply -f db-svc.yaml
```

```
ubuntu@reyes-nube:~$ kubectl get svc
NAME          TYPE      CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
db-svc        ClusterIP  10.43.65.190   <none>        3306/TCP   4d19h
kb-prod       ClusterIP  10.43.128.113 <none>        3000/TCP   49d
metabase-service ClusterIP  10.43.178.147 <none>        3000/TCP   4d19h
```

Luego hice los archivos de configuración YAML para desplegar y gestionar la aplicación Metabase en un clúster de Kubernetes. Para ello hice y ejecuté el **metabase-secret.sh** para que genere el **metabase-secret.yaml** . Este archivo crea un objeto Secret en Kubernetes llamado **metabase-secret** que almacena las credenciales y la configuración relacionada con la base de datos para la aplicación Metabase.

Para desplegar el recurso

```
ubuntu@reyes-nube ~$ kubectl apply -f metabase-secret.yaml
```

```
ubuntu@reyes-nube:~$ kubectl get secret
NAME          TYPE   DATA   AGE
db-secret     Opaque 3       5d15h
metabase-secret Opaque 4       4d18h
```

Después hice el **metabase-deploy.yaml** . Este archivo crea un objeto Deployment llamado **metabase**, que es utilizado para desplegar y gestionar la aplicación Metabase en el clúster de Kubernetes.

El **metabase-secret** proporciona las credenciales y configuración seguras necesarias para que los Pods dentro del **metabase** Deployment puedan acceder y autenticarse correctamente en la base de datos y configurar la aplicación Metabase adecuadamente. La relación entre ambos se establece a través de las variables de entorno en el archivo de configuración del Deployment.

Para desplegar el recurso

```
ubuntu@reyes-nube ~$ kubectl apply -f metabase-deploy.yaml
```

```
ubuntu@reyes-nube:~$ kubectl get deploy
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
kb-prod   1/1     1             1           49d
metabase  1/1     1             1           3d20h
```

Y por último hice el **metabase-svc.yaml** . Este archivo crea un objeto Service llamado **metabase-service**, que permite exponer la aplicación Metabase dentro del clúster o en el exterior.

El **metabase-service** proporciona una forma de acceder a los Pods individuales del **metabase** Deployment que ejecutan la aplicación Metabase. Permite que los usuarios y otras aplicaciones se comuniquen con la aplicación a través del servicio sin preocuparse por la ubicación exacta de los Pods.

Para desplegar el recurso

```
ubuntu@reyes-nube ~$ kubectl apply -f metabase-svc.yaml
```

```
ubuntu@reyes-nube:~$ kubectl get svc
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
db-svc        ClusterIP   10.43.65.190  <none>        3306/TCP  4d19h
kb-prod       ClusterIP   10.43.128.113 <none>        3000/TCP  49d
metabase-service ClusterIP   10.43.178.147 <none>        3000/TCP  4d19h
```

Finalmente hice el **web-ingress.yaml** . Este archivo de configuración YAML define un objeto Ingress en Kubernetes llamado **metabase-ingress** que configura el enrutamiento de tráfico desde el exterior hacia el servicio de la aplicación Metabase, es decir hacia el **metabase-service**.

Para desplegar el recurso

```
ubuntu@reyes-nube ~$ kubectl apply -f web-ingress.yaml
```

```
ubuntu@reyes-nube:~$ kubectl get ingress
NAME          CLASS   HOSTS
kb-ing        <none>  martingreyescaso3.my.kube.um.edu.ar
metabase-ingress <none>  martingreyescaso3.my.kube.um.edu.ar
```

NAME	CLASS	HOSTS	ADDRESS	PORTS	AGE
kb-ing	<none>	martingreyescaso3.my.kube.um.edu.ar	10.200.64.101,10.200.65.129,10.200.65.71,10.200.65.95,10.200.66.14,10.200.66.68,10.200.67.206,10.200.67.28	80	5d13h
metabase-ingress	<none>	martingreyescaso3.my.kube.um.edu.ar	10.200.64.101,10.200.65.129,10.200.65.71,10.200.65.95,10.200.66.14,10.200.66.68,10.200.67.206,10.200.67.28	80	3d20h

Ingreso a metabase desde el navegador:

<https://martingreyesdevcaso3.my.kube.um.edu.ar/>

The screenshot shows a web browser window with the URL `martingreyesdevcaso3.my.kube.um.edu.ar`. The page has a navigation bar with links like 'UM', 'Portal Estudiante', 'Cátedras', 'Gmail', 'GPT', 'OpenStack', 'GitHub', 'DockerHub', 'Economía de Mendoza', 'Play Console', 'YouTube', 'MeLI', 'Cursos', 'Calculadoras', 'Utilidades', 'Entretenimiento', and 'etc'. Below the navigation bar is a grid of icons. The main content area shows a registration form with a green checkmark and the text 'Tu idioma es Spanish'. The form is titled '¿Cómo deberíamos llamarte?' and includes fields for 'Nombre' (Martin), 'Apellidos' (Reyes), 'Email' (tinchoreyes123@gmail.com), 'Nombre empresa o equipo' (UM), 'Crea una contraseña' (tele2023), and 'Confirma tu contraseña' (tele2023). A 'Siguiente' button is at the bottom.

Tu idioma es Spanish

2 ¿Cómo deberíamos llamarte?

Nombre Martin Apellidos Reyes

Email tinchoreyes123@gmail.com

Nombre empresa o equipo UM

Crea una contraseña tele2023

Confirma tu contraseña tele2023

Siguiente

The screenshot shows the same web browser window, but the page has advanced to the next step. It shows a green checkmark and the text 'Hola, Martín. Encantado de conocerte!'. The form is titled 'Añade tus datos' and includes a dropdown for 'MySQL', a text field for 'Nombre para mostrar' (Dataset Google Mobility), a text field for 'Host' (db-svc), a text field for 'Port' (3306), a text field for 'Database name' (datos), a text field for 'Username' (root), and a text field for 'Password' (tele). A 'Siguiente' button is at the bottom. A small tooltip in the bottom right corner says '¿Necesitas ayuda para conectarte?' and 'Consulte nuestra documentación para saber cómo conectar su base de datos paso a paso.'.

Tu idioma es Spanish

Hola, Martín. Encantado de conocerte!

3 Añade tus datos

¿Estás listo para comenzar a explorar tus datos? Añádelos a continuación.
¿No estás listo? Pasa directamente a jugar con nuestra base de datos de muestra.

MySQL

Nombre para mostrar Dataset Google Mobility

Host db-svc

Port 3306

Database name datos

Username root

Password tele

¿Necesitas ayuda para conectarte?
Consulte nuestra documentación para saber cómo conectar su base de datos paso a paso.



Dataset Google Mobility / Mobility

Date entre enero 1, 2020 diciembre 31, 2020 Sub Region 1 es Mendoza Province Sub Region 2 es Capital Department

Retail And Recreation Percent Change From Baseline Workplaces Percent Change From Baseline Parks Percent Change From Baseline Grocery And Pharmacy Percent Change From Baseline

