Certified Tech Developer
The Ultimate Degree

Infraestructura III

Desplegamos Wordpress con Helm

Objetivo

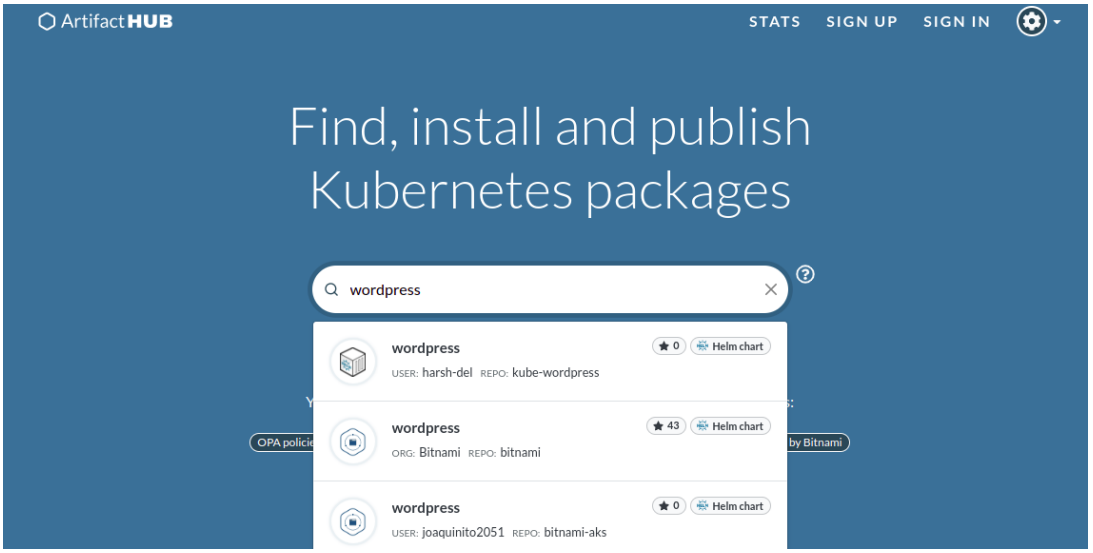
En este ejercicio, veremos cómo podemos desplegar pods de Wordpress con su correspondiente base de datos, personalizando algunos parámetros para cambiar las credenciales de admin, todo utilizando un chart de Helm.

¡Éxitos!

Buscamos información sobre el chart de Wordpress

Para ello vamos a buscar el chart en la página a continuación. Para el ejemplo seleccionaremos el repo de Bitnami

https://artifacthub.io/



Seleccionamos la segunda opción que aparece y nos mostrará toda la información acerca del paquete de Worpress de Bitnami. Le damos clic al botón **install** y nos mostrará los comandos para hacer la instalación.

Estos son los comandos:

Para agregar el repositorio de bitnami:

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

La salida del comando debe ser esta:

```
"bitnami" has been added to your repositories
```

Helm install: instalamos el paquete

Ejecutamos el siguiente comando para instalar el paquete de Wordpress de Bitnami (mi-wordpress va a ser el nombre que tenga el pod, podemos reemplazarlo por el nombre que queramos):

```
helm install my-wordpress oci://registry-1.docker.io/bitnamicharts/wordpress
```

La salida del comando nos mostrará bastante información. ¿Qué datos importantes nos muestra?

```
NAME: mi-wordpress
LAST DEPLOYED: Sun Jul 31 10:13:52 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: wordpress
CHART VERSION: 17.1.6
```


```
APP VERSION: 6.0.1

** Please be patient while the chart is being deployed **

Your WordPress site can be accessed through the following DNS name from
within your cluster:

    mi-wordpress.default.svc.cluster.local (port 80)

To access your WordPress site from outside the cluster follow the steps
below:

1. Get the WordPress URL by running these commands:

  NOTE: It may take a few minutes for the LoadBalancer IP to be
available.
        Watch the status with: 'kubectl get svc --namespace default -w
mi-wordpress'

   export SERVICE_IP=$(kubectl get svc --namespace default mi-wordpress
--include "{{ range (index .status.loadBalancer.ingress 0) }}{{ . }}{{
end }}")
   echo "WordPress URL: http://$SERVICE_IP/"
   echo "WordPress Admin URL: http://$SERVICE_IP/admin"

2. Open a browser and access WordPress using the obtained URL.

3. Login with the following credentials below to see your blog:

  echo Username: user
  echo Password: $(kubectl get secret --namespace default mi-wordpress
-o jsonpath="{.data.wordpress-password}" | base64 -d)
```

Vamos a validar que se haya creado todo lo necesario. Para ello vamos a listar todos los objetos del namespace Default donde estamos trabajando:

Para ello ejecutamos el comando siguiente:

```
kubectl get pods
```

La salida del comando debería ser similar a esta:

```
NAME                                READY   STATUS    RESTARTS   AGE
pod/mi-wordpress-659455d4bf-8mb68   1/1     Running   0          68s
pod/mi-wordpress-mariadb-0          1/1     Running   0          68s

NAME                            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                      AGE
service/kubernetes              ClusterIP      10.96.0.1       <none>        443/TCP                      18d
service/mi-wordpress            LoadBalancer   10.110.45.76    <pending>     80:31611/TCP,443:31940/TCP   68s
service/mi-wordpress-mariadb    ClusterIP      10.102.7.36     <none>        3306/TCP                     68s


NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mi-wordpress   1/1     1            1           68s

NAME                                      DESIRED   CURRENT   READY   AGE
replicaset.apps/mi-wordpress-659455d4bf   1         1         1       68s

NAME                                    READY   AGE
statefulset.apps/mi-wordpress-mariadb   1/1     68s
```

Ahí podemos ver que Helm nos ha desplegado el chart de Wordpress, el cual levantó dos pods: uno para el frontend y otro para el backend. Por el nombre “pod/mi-wordpress-mariadb-0”, podemos inferir que este pod contiene una base de datos de mariadb. A su vez, vemos que se crearon services con el nombre mi-wordpress, un deployment con el mismo nombre con su replicaset y un statefulset para la base de datos. ¡Todo esto con una sola línea de código de Helm!

Mirando en detalle los services, podemos ver que para la base de datos hay un service tipo ClusterIP y para el frontend un service tipo LoadBalancer. Este último nos permitirá accederlo desde fuera del cluster.

Pero en la columna EXTERNAL-IP vemos que está pending. Esto ocurre porque este ejercicio se está ejecutando en Minikube. Para lograr que el service LoadBalancer tenga una IP external se debe correr **en una terminal nueva** el siguiente comando:

```
minikube tunnel
```

NOTA: es importante dejar este comando corriendo en una terminal aparte mientras probamos acceder al servicio.

Si estamos en docker desktop deberemos hacer un tunel al pod para acceder (reemplazando el nombre del pod por el que ustedes tengan):

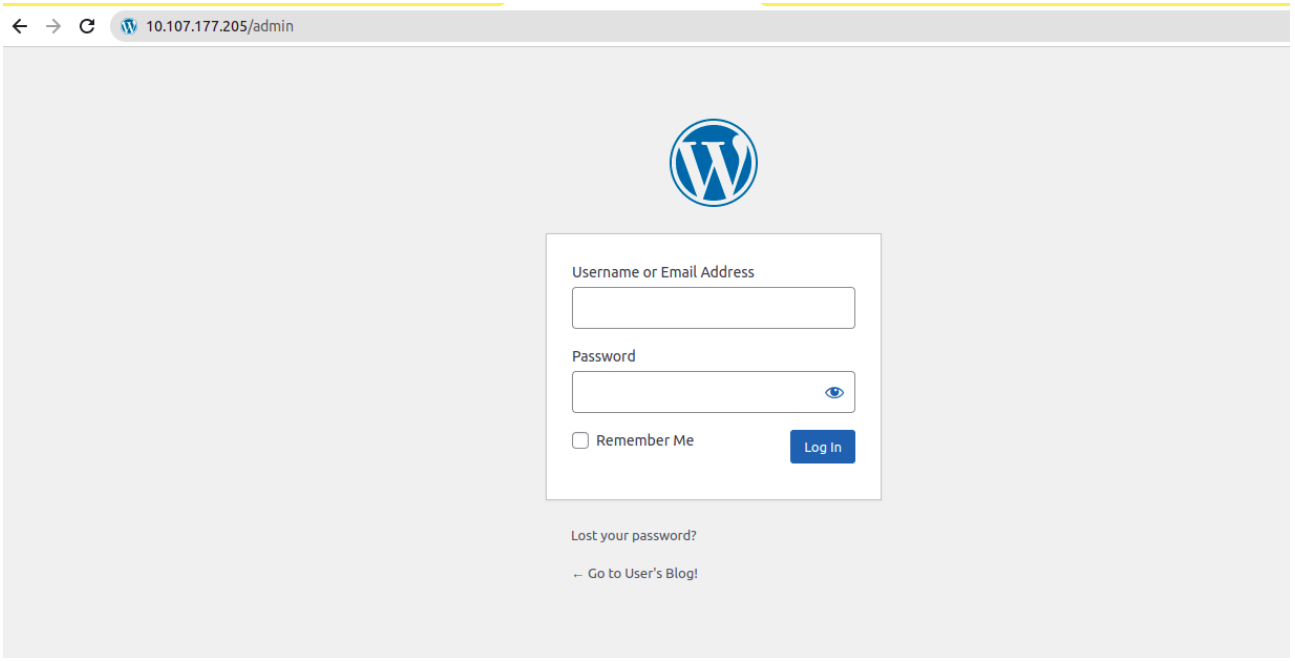
```
kubectl port-forward pod/mi-wordpress-566c5d949-z2ndj 4000:8080
```

y accederemos desde **localhost:4000/admin**

Volvemos a la terminal en la que estábamos y con el comando debajo vemos que el service LoadBalancer ya tiene una IP externa:

```
maury@S1-MD:~$ kubectl get service mi-wordpress
NAME           TYPE           CLUSTER-IP       EXTERNAL-IP      PORT(S)                      AGE
mi-wordpress   LoadBalancer   10.107.177.205   10.107.177.205   80:31439/TCP,443:30164/TCP   131m
```

Ahora podemos acceder desde el navegador. Para ello copiamos la IP que figura en la columna EXTERNAL-IP y en el navegador ponemos la dirección de este modo:



Podemos obtener el usuario y la contraseña copiando los comandos de las instrucciones que nos mostró el chart luego de hacer el install; serían estos:

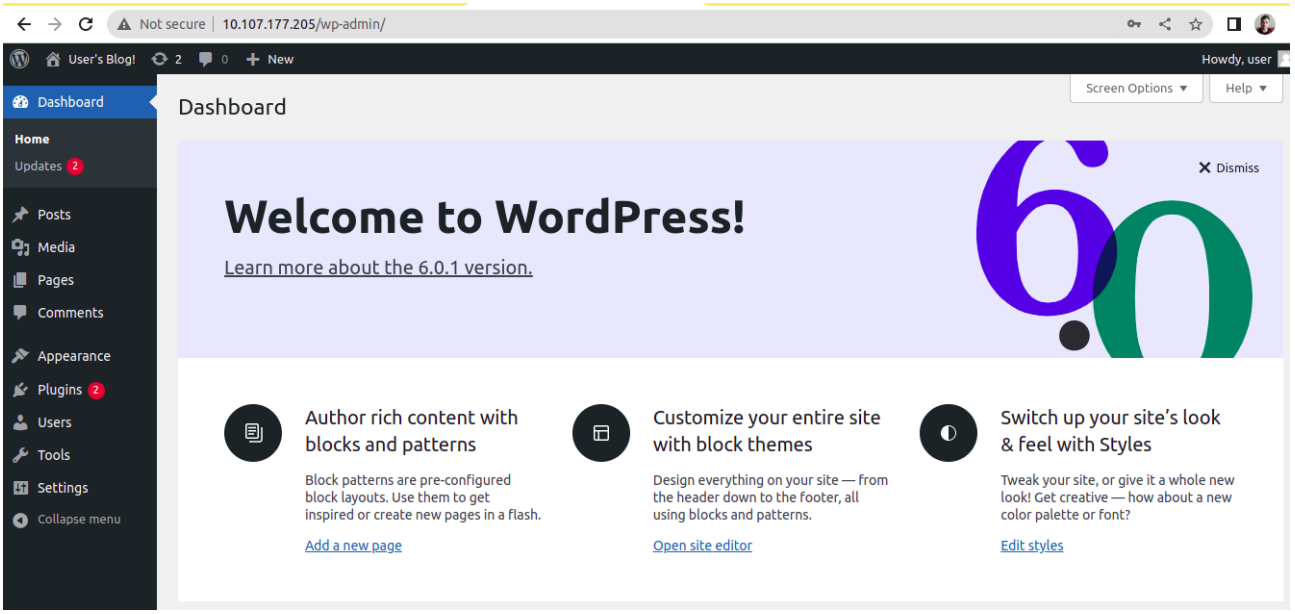
```
3. Login with the following credentials below to see your blog:

  echo Username: user
  echo Password: $(kubectl get secret --namespace default mi-wordpress
-o jsonpath="{.data.wordpress-password}" | base64 -d)
```

El primer comando nos devuelve el user, que se llama **user.**
El segundo nos dará el password:

```
maury@S1-MD:~$ echo Password: $(kubectl get secret --namespace default
mi-wordpress -o jsonpath="{.data.wordpress-password}" | base64 -d)
Password: TB2iXe7zsp
```

Copiamos el password que aparece en la tercera línea y nos logueamos:



Nos debe aparecer la consola de Wordpress. ¡Con esto finalizamos el ejercicio!