Terraform-Módulo 2: Chart de Helm

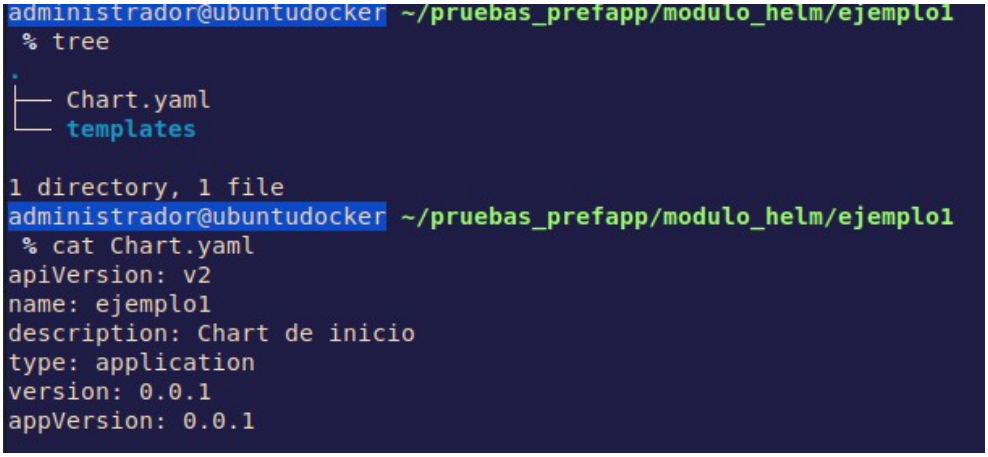
Manuel Vergara

Taula de continguts

1 De los valores y su interpolación

1.1 De los valores y su interpolación

```
mkdir ejemplo1

cd ejemplo1

curl https://raw.githubusercontent.com/frmadem/helm-curso/master/charts/
inicial/starter.sh | NOMBRE=ejemplo1 bash -x
```



Creamos la plantilla del pod en templates

```
# templates/pod.yaml

apiVersion: v1
kind: Pod
metadata:
  name: web
spec:
  containers:
    - name: web
      image: nginx
      ports:
        - name: web
          containerPort: 80
          protocol: TCP
```

Ahora renderizamos toda la chart en un manifiesto único

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/ejemplo1
 % helm template .
---
# Source: ejemplo1/templates/pod.yaml
# templates/pod.yaml

apiVersion: v1
kind: Pod
metadata:
  name: web
spec:
  containers:
    - name: web
      image: nginx
      ports:
        - name: web
          containerPort: 80
          protocol: TCP
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/ejemplo1
```

1.2 Agregando valores a nuestra chart

Creamos el fichero values.yaml

```
# ejemplo1/values.yaml
imagen: nginx
```

Y cambio el valor de image de la plantilla para que coja values.yaml

```
image: {{ .Values.imagen }}
```

Ejecuto de nuevo el template y tenemos el mismo resultado, obteniendo la imagen nginx de la renderización.

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/ejemplo1
 % vim values.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/ejemplo1
 % vim templates/pod.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/ejemplo1
 % helm template .
---
# Source: ejemplo1/templates/pod.yaml
# templates/pod.yaml

apiVersion: v1
kind: Pod
metadata:
  name: web
spec:
  containers:
    - name: web
      image: nginx
      ports:
        - name: web
          containerPort: 80
          protocol: TCP
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/ejemplo1
```

1.3 Sobreescribiendo los valores por defecto

Cambio la versión de nginx en mis_valores.yaml

```
# ejemplo1/mis_valores.yaml
imagen: nginx:1.19.1
~
```

Y piso values.yaml con mis_valores.yaml

```
 % cp values.yaml mis_valores.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/ejemplo1
 % vim mis_valores.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/ejemplo1
 % helm template . -f mis_valores.yaml
---
# Source: ejemplo1/templates/pod.yaml
# templates/pod.yaml

apiVersion: v1
kind: Pod
metadata:
  name: web
spec:
  containers:
    - name: web
      image: nginx:1.19.1
      ports:
        - name: web
          containerPort: 80
          protocol: TCP
```

2 De la creación de charts

2.1 Helm create

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm create mi-primera-chart
Creating mi-primera-chart
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % ls
ejemplo1  mi-primera-chart  values.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % tree mi-primera-chart
mi-primera-chart
├── charts
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── hpa.yaml
│   ├── ingress.yaml
│   ├── NOTES.txt
│   ├── serviceaccount.yaml
│   ├── service.yaml
│   └── tests
│       └── test-connection.yaml
└── values.yaml

3 directories, 10 files
```

2.2 Instalando nuestra Chart

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k create ns primera-chart
namespace/primera-chart created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm install mi-primera-chart mi-primera-chart -n primera-chart
NAME: mi-primera-chart
LAST DEPLOYED: Wed Nov  2 12:09:40 2022
NAMESPACE: primera-chart
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace primera-chart -l "app.kubernetes.io/name=mi-primera-chart,app.kubernetes.io/instance=mi-primera-chart"
o jsonpath="{.items[0].metadata.name}")
  export CONTAINER_PORT=$(kubectl get pod --namespace primera-chart $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl --namespace primera-chart port-forward $POD_NAME 8080:$CONTAINER_PORT
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm list -A
NAME                NAMESPACE       REVISION        UPDATED                                   STATUS          CHART                   APP VERSION
mi-primera-chart    primera-chart   1               2022-11-02 12:09:40.420740819 +0100 CET deployed        mi-primera-chart-0.1.0  1.16.0
```

```
 % k get all -n primera-chart
NAME                                    READY   STATUS    RESTARTS   AGE
pod/mi-primera-chart-cd8c798c9-tgpwd    1/1     Running   0          46s

NAME                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
service/mi-primera-chart  ClusterIP   10.96.101.95   <none>        80/TCP    46s

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mi-primera-chart  1/1     1            1           46s

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/mi-primera-chart-cd8c798c9   1         1         1       46s
```

2.3 Adaptando el proyecto por defecto

Preparamos la estructura dejando solo un yaml de pod nginx

```
 % rm -rf templates/* && touch templates/NOTES.txt && tree
zsh: sure you want to delete the only file in /home/administrador/pruebas_prefapp/modulo_helm/mi-primera-chart/templates [yn]? y
.
├── charts
├── Chart.yaml
├── templates
│   └── NOTES.txt
└── values.yaml

2 directories, 3 files
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/mi-primera-chart
 % cat /dev/null > values.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/mi-primera-chart
 % vim templates/pod.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/mi-primera-chart
 % tree
.
├── charts
├── Chart.yaml
├── templates
│   ├── NOTES.txt
│   └── pod.yaml
└── values.yaml

2 directories, 4 files
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/mi-primera-chart
 %
```

Añado el texto que se mostrará con helm install en NOTES.txt

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/mi-primera-chart
 % echo "Puedes conseguir la URL de la aplicación con este comando:
  kubectl --namespace {{ .Release.Namespace }} port-forward pod/web-nginx 8080:80" >> templates/NOTES.txt
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/mi-primera-chart
 %
```

2.4 Finalizar y distribuír nuestra Chart

Añado values y el pod quedando así:

values.yaml

```
imagen:
  nombre: nginx
  tag: ""
```

```
#templates/pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: web-nginx
spec:
  containers:
    - name: web
      image: "{{ .Values.imagen.nombre }}:{{ .Values.imagen.tag | default .Chart.AppVersion }}"
      ports:
        - name: web
          containerPort: 80
          protocol: TCP
```

Creo el paquete

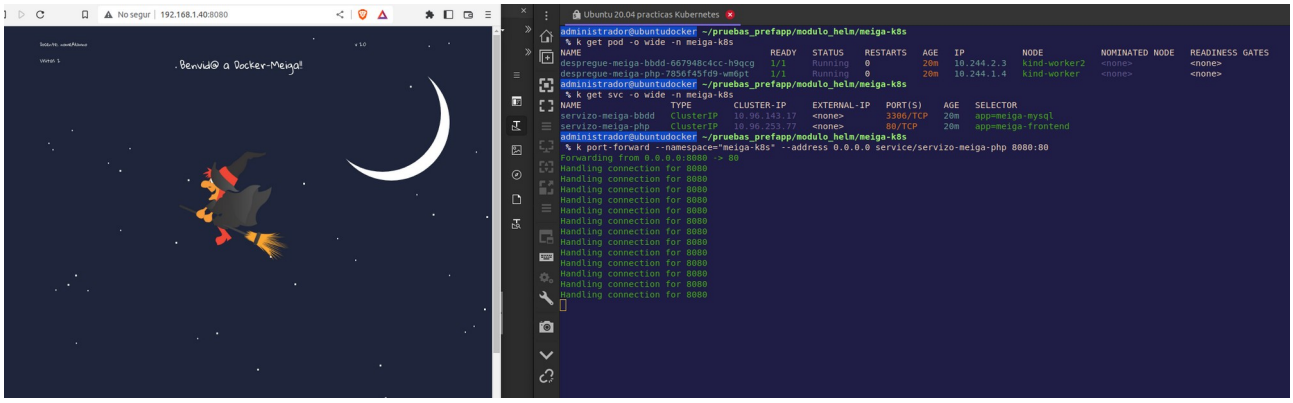
```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/mi-primera-chart
 % helm template .
---
# Source: mi-primera-chart/templates/pod.yaml
# templates/pod.yaml

apiVersion: v1
kind: Pod
metadata:
  name: web-nginx
spec:
  containers:
    - name: web
      image: "nginx:1.16.0"
      ports:
        - name: web
          containerPort: 80
          protocol: TCP
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/mi-primera-chart
 % helm package .
Successfully packaged chart and saved it to: /home/administrador/pruebas_prefapp/modulo_helm/mi-primera-chart/mi-prim
era-chart-0.1.0.tgz
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/mi-primera-chart
 % ls
charts  Chart.yaml  mi-primera-chart-0.1.0.tgz  templates  values.yaml
```

3 Práctica guiada: Proyecto Meiga en Helm

Comprobamos que funcionan todos los artefactos en k8s. Los lanzo

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k create ns meiga-k8s
namespace/meiga-k8s created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k get ns
NAME                   STATUS   AGE
default                Active   6d17h
ingress-nginx          Active   6d17h
kube-node-lease        Active   2m51s
kube-public            Active   6d17h
kube-system            Active   6d17h
local-path-storage     Active   6d17h
meiga-k8s              Active   3s
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k apply -f configmap.yaml -n meiga-k8s
configmap/meiga-config created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k apply -f secret.yaml -n meiga-k8s
secret/meiga-secrets created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k apply -f frontend-deploy.yaml -n meiga-k8s
deployment.apps/despregue-meiga-php created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k apply -f bbdd-deploy.yaml -n meiga-k8s
deployment.apps/despregue-meiga-bbdd created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k apply -f frontend-service.yaml -n meiga-k8s
service/servizo-meiga-php created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k apply -f bbdd-service.yaml -n meiga-k8s
service/servizo-meiga-bbdd created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k get all -n meiga-k8s
NAME                                       READY   STATUS    RESTARTS   AGE
pod/despregue-meiga-bbdd-667948c4cc-h9qcg  1/1     Running   0          34s
pod/despregue-meiga-php-7856f45fd9-wm6pt   1/1     Running   0          46s

NAME                       TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
service/servizo-meiga-bbdd ClusterIP   10.96.143.17   <none>        3306/TCP   16s
service/servizo-meiga-php  ClusterIP   10.96.253.77   <none>        80/TCP     24s

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/despregue-meiga-bbdd   1/1     1            1           34s
deployment.apps/despregue-meiga-php    1/1     1            1           46s

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/despregue-meiga-bbdd-667948c4cc   1         1         1       34s
replicaset.apps/despregue-meiga-php-7856f45fd9    1         1         1       46s
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
```

Compruebo



Lo borro todo

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k delete --all service -n meiga-k8s
service "servizo-meiga-bbdd" deleted
service "servizo-meiga-php" deleted
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k delete --all deploy -n meiga-k8s
deployment.apps "despregue-meiga-bbdd" deleted
deployment.apps "despregue-meiga-php" deleted
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k delete --all secret -n meiga-k8s
secret "meiga-secrets" deleted
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k delete --all configmap -n meiga-k8s
configmap "kube-root-ca.crt" deleted
configmap "meiga-config" deleted
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 % k get all -n meiga-k8s
No resources found in meiga-k8s namespace.
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-k8s
 %
```

3.1 Probando Helm

Creamos el proyecto y revisamos Chart.yaml

```
 % helm create meiga-project
Creating meiga-project
administrador@ubuntudocker ~/pruebas_
 % vim meiga-project/Chart.yaml
```

```
apiVersion: v2
name: meiga-project
description: A Helm chart for Kubernetes
type: application
version: 0.1.0
appVersion: "1.16.0"
~
```

Borramos el contenido de values.yaml

```
dministrador@ubuntudocker ~/pruebas_prefa
% : > meiga-project/values.yaml
dministrador@ubuntudocker ~/pruebas_prefa
% cat meiga-project/values.yaml
dministrador@ubuntudocker ~/pruebas_prefa
```

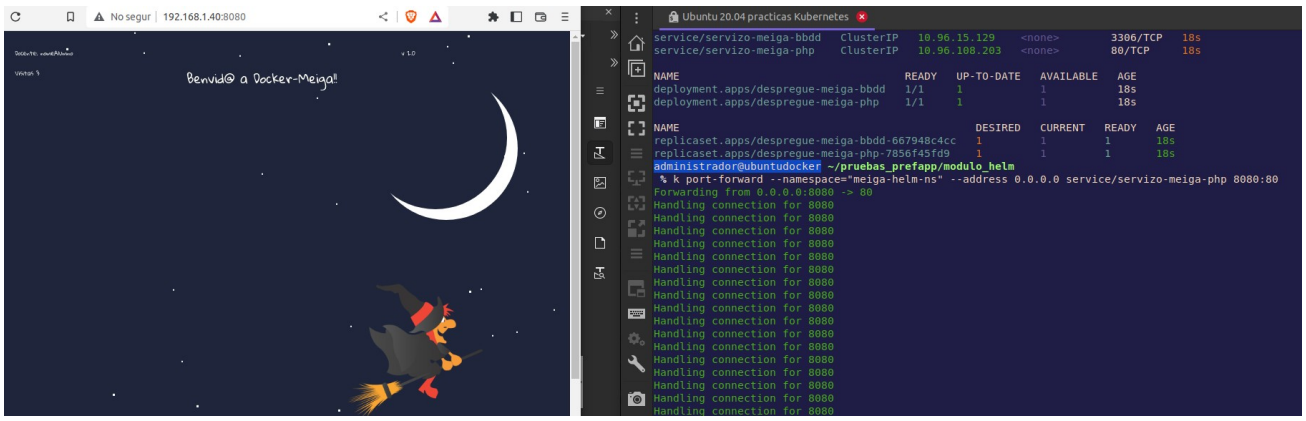
Borramos el contenido de templates y añadimos el proyecto meigas

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % rm -rf meiga-project/templates/*
zsh: sure you want to delete all the files in /home/administrador/pruebas_prefapp/modulo_helm/meiga-project/templates [yn]? y
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % tree meiga-project
meiga-project
├── charts
├── Chart.yaml
├── templates
└── values.yaml

2 directories, 2 files
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % cp meiga-k8s/* meiga-project/templates/.
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % tree meiga-project
meiga-project
├── charts
├── Chart.yaml
├── templates
│   ├── bbdd-deploy.yaml
│   ├── bbdd-service.yaml
│   ├── configmap.yaml
│   ├── frontend-deploy.yaml
│   ├── frontend-service.yaml
│   └── secret.yaml
└── values.yaml

2 directories, 8 files
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```

Ahora creamos la release

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k create ns meiga-helm-ns
namespace/meiga-helm-ns created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k get ns
NAME                 STATUS   AGE
default              Active   6d18h
ingress-nginx        Active   6d18h
kube-node-lease      Active   42m
kube-public          Active   6d18h
kube-system          Active   6d18h
local-path-storage   Active   6d18h
meiga-helm-ns        Active   4s
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm install meiga-release meiga-project -n meiga-helm-ns
NAME: meiga-release
LAST DEPLOYED: Thu Nov  3 11:18:06 2022
NAMESPACE: meiga-helm-ns
STATUS: deployed
REVISION: 1
TEST SUITE: None
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k get all -n meiga-helm-ns
NAME                                        READY   STATUS    RESTARTS   AGE
pod/despregue-meiga-bbdd-667948c4cc-6mqfv   1/1     Running   0          18s
pod/despregue-meiga-php-7856f45fd9-fwxsc    1/1     Running   0          18s

NAME                          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/servizo-meiga-bbdd    ClusterIP   10.96.15.129    <none>        3306/TCP   18s
service/servizo-meiga-php     ClusterIP   10.96.108.203   <none>        80/TCP     18s

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/despregue-meiga-bbdd   1/1     1            1           18s
deployment.apps/despregue-meiga-php    1/1     1            1           18s

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/despregue-meiga-bbdd-667948c4cc   1         1         1       18s
replicaset.apps/despregue-meiga-php-7856f45fd9    1         1         1       18s
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```

Y comprobamos



También compruebo que puedo listar el release y ver más detalles con «get»

```
% helm list -n meiga-helm-ns
NAME            NAMESPACE       REVISION        UPDATED                                   STATUS          CHART                   APP VERSIO
meiga-release   meiga-helm-ns   1               2022-11-03 11:18:06.416590119 +0100 CET deployed        meiga-project-0.1.0     1.16.0
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
% helm get all meiga-release -n meiga-helm-ns
NAME: meiga-release
LAST DEPLOYED: Thu Nov  3 11:18:06 2022
NAMESPACE: meiga-helm-ns
STATUS: deployed
REVISION: 1
TEST SUITE: None
USER-SUPPLIED VALUES:
null

COMPUTED VALUES:
{}

HOOKS:
MANIFEST:
---
# Source: meiga-project/templates/secret.yaml
# secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: meiga-secrets
type: Opaque
data: # aquí van os datos
  root-password: Y29udHJhc2luYWw=
---
# Source: meiga-project/templates/configmap.yaml
# configmap.yaml
#
apiVersion: v1
kind: ConfigMap # o tipo de artefacto
metadata:
  name: meiga-config # ten un nome
  labels:
    tipo: "configuracions"  # podemoslle meter labels
data:
  CURSO: "nomeCurso"
  DOCENTE: "nomeAlumno"
  MYSQL_HOST: servizo-meiga-bbdd
  MYSQL_USER: "root"
  MYSQL_DATABASE: "meiga"
---
# Source: meiga-project/templates/bbdd-service.yaml
# bbdd-service.yaml

kind: Service
apiVersion: v1
metadata:  # esta é a parte de identificación do servizo
  name: servizo-meiga-bbdd
```

Desinstalo la release.

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
% helm uninstall meiga-release -n meiga-helm-ns
release "meiga-release" uninstalled
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
%
```

3.2 Renombrando nuestro artefactos

despregue-meiga-bbdd --> despregue-{{ .Release.Name }}-bbdd

despregue-meiga-php --> despregue-{{ .Release.Name }}-php

servizo-meiga-php --> servizo-{{ .Release.Name }}-php

servizo-meiga-bbdd --> servizo-{{ .Release.Name }}-bbdd

meiga-secrets --> {{ .Release.Name }}-secrets

meiga-config --> {{ .Release.Name }}-config

3.3 Configurando values.yaml

- Cambio el puerto del fronted
- Imágenes como parámetro
- Migrar los valores de nuestro configmap a Values
- Contenido del secrets

3.3.1 Comprobaciones

El archivo values.yaml ha quedado así:

```
# meiga-project/values.yaml
puertos:
  servicio:
    fronted: 80
imagenes:
  frontend: elberto/meiga-php-lite
  bbdd: mysql:5.7
## Variables de env que modifican fronted
env:
  CURSO: "Aprende Helm"
  DOCENTE: "Estudiante1"
mysql:
  user: "root"
  database: "meiga"
secretos:
  rootpass: contrasinal
```

Hago una comprobación con lint para ver si todo está correcto y renderizo en un manifiesto para comprobar

```
 % helm lint .
==> Linting .
[INFO] Chart.yaml: icon is recommended

1 chart(s) linted, 0 chart(s) failed
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-project
 % helm template .
---
# Source: meiga-project/templates/secret.yaml
# secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: RELEASE-NAME-secrets
type: Opaque
data: # aquí van os datos
root-password: Y29udHJhc2luYWw=
---
# Source: meiga-project/templates/configmap.yaml
# configmap.yaml
#
apiVersion: v1
kind: ConfigMap
metadata:
  name: RELEASE-NAME-config
  labels:
    tipo: "configuracions"  # podemoslle meter labels
data:
  MYSQL_HOST: servizo-RELEASE-NAME-bbdd
  MYSQL_USER: root
  MYSQL_DATABASE: meiga

  CURSO: Aprende Helm
  DOCENTE: Estudiante1
---
# Source: meiga-project/templates/bbdd-service.yaml
# bbdd-service.yaml

kind: Service
apiVersion: v1
metadata:
  name: servizo-RELEASE-NAME-bbdd
spec:
  selector:   # esta é a parte de selección
    app: meiga-mysql
  ports:  # esta é a parte de especificación propia
  - protocol: TCP
    port: 3306
    targetPort: 3306
---
# Source: meiga-project/templates/frontend-service.yaml
```

Instalo cambiando sobre la marcha el env de la bbdd docente

```
NAME: meiga-realease
LAST DEPLOYED: Thu Nov  3 13:01:09 2022
NAMESPACE: meiga-helm-ns
STATUS: deployed
REVISION: 1
TEST SUITE: None
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm list
NAME    NAMESPACE       REVISION        UPDATED STATUS  CHART   APP VERSION
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm list --all
NAME    NAMESPACE       REVISION        UPDATED STATUS  CHART   APP VERSION
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm list -A
NAME                    NAMESPACE       REVISION        UPDATED                                   STATUS          CHART                   APP VERSION
meiga-pro-v2            meiga-helm-ns   1               2022-11-03 12:56:42.794655765 +0100 CET failed          meiga-project-0.1.0     1.16.0
meiga-realease          meiga-helm-ns   1               2022-11-03 13:01:09.868146594 +0100 CET deployed        meiga-project-0.1.0     1.16.0
meiga-release           meiga-helm-ns   1               2022-11-03 12:45:56.790938852 +0100 CET failed          meiga-project-0.1.0     1.16.0
meiga-release-v2        meiga-helm-ns   1               2022-11-03 12:48:49.064210398 +0100 CET failed          meiga-project-0.1.0     1.16.0
meiga-v2                meiga-helm-ns   1               2022-11-03 12:54:53.258820322 +0100 CET failed          meiga-project-0.1.0     1.16.0
meiga2                  meiga-helm-ns   1               2022-11-03 12:49:20.990506691 +0100 CET failed          meiga-project-0.1.0     1.16.0
```

Desinstalo e instalo de nuevo cambiando el nombre en my-values.yaml

meiga-project/my-values.yaml

env:

  DOCENTE: "Pepito"

Me encuentro el siguiente error cuando intento instalar de nuevo cambiando parámetros:

```
 % helm install meiga-realease meiga-project/ -f meiga-project/my-values.yaml -n meiga-helm-ns
Error: INSTALLATION FAILED: cannot re-use a name that is still in use
```

Según he encontrado en foros https://github.com/helm/helm/issues/7527, es porque helm protege el namespace para que no se guarde con el mismo nombre la release. Para evitar que choquen, como se comenta aquí: https://prefapp.github.io/formacion/cursos/helm/#/./02_helm_charts/03_practica_guiada_meiga?id=renombrando-nuestro-artefactos

La solución que proponen en el foro para utilizar el mismo namespaces es borrarlo y crearlo de nuevo

```
 % helm install meiga-realease meiga-project/ -f meiga-project/my-values.yaml -n meiga-helm-ns
Error: INSTALLATION FAILED: cannot re-use a name that is still in use
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k get ns meiga-helm-ns
NAME            STATUS   AGE
meiga-helm-ns   Active   114m
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k delete ns meiga-helm-ns
namespace "meiga-helm-ns" deleted
^[[A^[[D%
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k create ns meiga-helm-ns
namespace/meiga-helm-ns created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm install meiga-realease meiga-project/ -f meiga-project/my-values.yaml -n meiga-helm-ns
NAME: meiga-realease
LAST DEPLOYED: Thu Nov  3 13:13:02 2022
NAMESPACE: meiga-helm-ns
STATUS: deployed
REVISION: 1
TEST SUITE: None
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm list -n meiga-helm-ns
NAME            NAMESPACE       REVISION        UPDATED                                   STATUS          CHART                   APP VERSION
meiga-realease  meiga-helm-ns   1               2022-11-03 13:13:02.181952493 +0100 CET deployed        meiga-project-0.1.0     1.16.0
```

4 Continuación de la Práctica Guiada: Mejorando nuestra Meiga

4.1 Cambiando la BBDD a una sub-Chart

Borramos los manifiestos del servicio y del deploy de bbdd de la carpeta templates.

```
% cp -r meiga-project meiga-project-v2
dministrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
% rm -rf meiga-project-v2/templates/bbdd-*
dministrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```

Añadimos a Chart.yaml la dependencia:

```
#nueva dependencia
dependencies:
- name: mysql
 version: "8.3.1"
 repository: https://charts.bitnami.com/bitnami
```

Añadimos la configuración de la subchart a values.yaml

```
#SUBCHART mysql
mysql:
 image:
   tag: 5.7
 auth:
   rootPassword: contrasinal
   database: "meiga"
```

Editamos configmap para apuntar al nuevo servicio

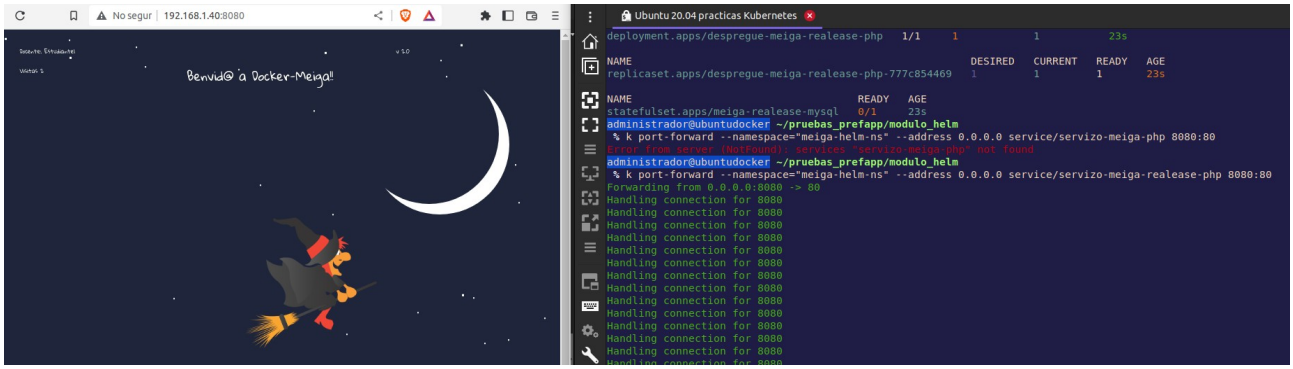
```
...
data:
  MYSQL_HOST: {{ .Release.Name }}-mysql
...
```

Descargo la dependencia, es decir, el repo de mysql

```
% helm dependency build
Getting updates for unmanaged Helm repositories...
...Successfully got an update from the "https://charts.bitnami.com/bitnami" chart repository
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "metrics-server" chart repository
...Successfully got an update from the "hashicorp" chart repository
...Successfully got an update from the "artifact-hub" chart repository
...Successfully got an update from the "elastic" chart repository
...Unable to get an update from the "bitnami" chart repository (https://charts.bitnami.com):
        failed to fetch https://charts.bitnami.com/index.yaml : 403 Forbidden
...Successfully got an update from the "jenkins" chart repository
...Successfully got an update from the "datadog" chart repository
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. *Happy Helming!*
Saving 1 charts
Downloading mysql from repo https://charts.bitnami.com/bitnami
Deleting outdated charts
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-project-v2
```

y lo instalo

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm install meiga-realease meiga-project-v2/ -n meiga-helm-ns
NAME: meiga-realease
LAST DEPLOYED: Thu Nov  3 16:12:22 2022
NAMESPACE: meiga-helm-ns
STATUS: deployed
REVISION: 1
TEST SUITE: None
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k get all -n meiga-helm-ns
NAME                                                READY   STATUS    RESTARTS   AGE
pod/despregue-meiga-realease-php-777c854469-vclpn   1/1     Running   0          23s
pod/meiga-realease-mysql-0                          0/1     Running   0          23s

NAME                                  TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
service/meiga-realease-mysql          ClusterIP   10.96.179.66   <none>        3306/TCP   23s
service/meiga-realease-mysql-headless ClusterIP   None           <none>        3306/TCP   23s
service/servizo-meiga-realease-php    ClusterIP   10.96.232.69   <none>        80/TCP     23s

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/despregue-meiga-realease-php    1/1     1            1           23s

NAME                                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/despregue-meiga-realease-php-777c854469    1         1         1       23s

NAME                                  READY   AGE
statefulset.apps/meiga-realease-mysql 0/1     23s
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```



4.2 Añadir un Ingress para nuestro servicio

Hago una copia con v2 de todos los archivos. Añadimos la dependencia de ingress a Chart.yaml

```
dependencies:
- name: ingress-nginx
 version: "3.23.0"
 repository: https://kubernetes.github.io/ingress-nginx
```

Añadimos la configuración de ingress:

```
#values.yaml
...
ingress-nginx:
 controller:
   scope:
     enabled: true # defaults to .Release.Namespace
#Ajustes para Google Cloud
   service:
     internal:
       enabled: true
       annotations:
         # Create internal LB
```

```
        cloud.google.com/load-balancer-type: "Internal"
        # Any other annotation can be declared here.
```

Creo ingress.yaml redireccionando cualquier path al servidor meiga. Y descargo el nginx

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-project-v2
 % helm dependency update
Getting updates for unmanaged Helm repositories...
...Successfully got an update from the "https://kubernetes.github.io/ingress-nginx" chart repository
...Successfully got an update from the "https://charts.bitnami.com/bitnami" chart repository
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "metrics-server" chart repository
...Successfully got an update from the "hashicorp" chart repository
...Successfully got an update from the "elastic" chart repository
...Unable to get an update from the "bitnami" chart repository (https://charts.bitnami.com):
        failed to fetch https://charts.bitnami.com/index.yaml : 403 Forbidden
...Successfully got an update from the "artifact-hub" chart repository
...Successfully got an update from the "jenkins" chart repository
...Successfully got an update from the "datadog" chart repository
...Successfully got an update from the "prometheus-community" chart repository
Update Complete. *Happy Helming!*
Saving 2 charts
Downloading mysql from repo https://charts.bitnami.com/bitnami
Downloading ingress-nginx from repo https://kubernetes.github.io/ingress-nginx
Deleting outdated charts
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/meiga-project-v2
```

Instalo

```
 % k create ns meiga-project-v2-ns
namespace/meiga-project-v2-ns created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm install meiga-release meiga-project-v2 -n meiga-project-v2-ns
NAME: meiga-release
LAST DEPLOYED: Fri Nov  4 13:29:54 2022
NAMESPACE: meiga-project-v2-ns
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

Ahora mismo tengo dos release con el mismo nombre pero en diferentes namespaces

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm list -A
NAME            NAMESPACE             REVISION        UPDATED                                   STATUS          CHART                   APP VERSION
meiga-release   meiga-project-ns      1               2022-11-04 13:28:56.709109583 +0100 CET   deployed        meiga-project-0.1.0     1.16.0
meiga-release   meiga-project-v2-ns   1               2022-11-04 13:29:54.349648623 +0100 CET   deployed        meiga-project-0.1.0     1.16.0
```

Los artefactos de la primera versión

```
 % k get all -n meiga-project-ns
NAME                                                  READY   STATUS    RESTARTS   AGE
pod/despregue-meiga-release-bbdd-65596f8f-m88s9       1/1     Running   0          17m
pod/despregue-meiga-release-php-5c87c87df-24t82       1/1     Running   0          17m

NAME                                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/servizo-meiga-release-bbdd    ClusterIP   10.96.22.238    <none>        3306/TCP   17m
service/servizo-meiga-release-php     ClusterIP   10.96.103.195   <none>        80/TCP     17m

NAME                                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/despregue-meiga-release-bbdd     1/1     1            1           17m
deployment.apps/despregue-meiga-release-php      1/1     1            1           17m

NAME                                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/despregue-meiga-release-bbdd-65596f8f       1         1         1       17m
replicaset.apps/despregue-meiga-release-php-5c87c87df       1         1         1       17m
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```

Y los artefactos de la versión instalada con ingress

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k get all -n meiga-project-v2-ns
NAME                                                     READY   STATUS    RESTARTS   AGE
pod/despregue-meiga-release-php-5c87c87df-kdglg          1/1     Running   0          16m
pod/meiga-release-ingress-nginx-controller-dd68d7db6-8f4vs   1/1     Running   0          16m
pod/meiga-release-mysql-0                                1/1     Running   0          16m

NAME                                                 TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)                      AGE
service/meiga-release-ingress-nginx-controller       LoadBalancer   10.96.15.94    <pending>     80:31259/TCP,443:31596/TCP   16m
service/meiga-release-ingress-nginx-controller-admission   ClusterIP      10.96.61.29    <none>        443/TCP                      16m
service/meiga-release-ingress-nginx-controller-internal    LoadBalancer   10.96.227.39   <pending>     80:32683/TCP,443:30269/TCP   16m
service/meiga-release-mysql                          ClusterIP      10.96.21.0     <none>        3306/TCP                     16m
service/meiga-release-mysql-headless                 ClusterIP      None           <none>        3306/TCP                     16m
service/servizo-meiga-release-php                    ClusterIP      10.96.14.194   <none>        80/TCP                       16m

NAME                                                 READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/despregue-meiga-release-php          1/1     1            1           16m
deployment.apps/meiga-release-ingress-nginx-controller   1/1     1            1           16m

NAME                                                            DESIRED   CURRENT   READY   AGE
replicaset.apps/despregue-meiga-release-php-5c87c87df           1         1         1       16m
replicaset.apps/meiga-release-ingress-nginx-controller-dd68d7db6   1         1         1       16m

NAME                                 READY   AGE
statefulset.apps/meiga-release-mysql   1/1     16m
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 %
```

Este es el describe del ingress

```
 % k get ing --all-namespaces
NAMESPACE             NAME                    CLASS    HOSTS   ADDRESS   PORTS   AGE
meiga-project-v2-ns   meiga-release-ingress   <none>   *                 80      17m
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % k describe ing meiga-release-ingress -n meiga-project-v2-ns
Name:             meiga-release-ingress
Labels:           app.kubernetes.io/managed-by=Helm
Namespace:        meiga-project-v2-ns
Address:
Ingress Class:    <none>
Default backend:  <default>
Rules:
  Host        Path  Backends
  ----        ----  --------
  *
              /   servizo-meiga-release-php:80 (10.244.2.3:80)
Annotations:  kubernetes.io/ingress.class: nginx
              meta.helm.sh/release-name: meiga-release
              meta.helm.sh/release-namespace: meiga-project-v2-ns
Events:
  Type    Reason  Age   From                      Message
  ----    ------  ----  ----                      -------
  Normal  Sync    18m   nginx-ingress-controller  Scheduled for sync
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 %
```

No tengo dirección donde conectarme porque tendría que estar conectado a un proveedor cloud.

4.3 Crear un NOTES.txt

Guardo el documento:

```
#templates/NOTEST.txt
***INSTRUCCIONES***
  Puedes acceder a la web Meiga-php en la dirección proporcionada por el
ingress:
  $ kubectl describe ing {{ .Release.Name }}-ingress -n {{ .Release.Namespace }}
```

Y compruebo que cuando actualizas la release también sale el mensaje:

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm upgrade meiga-release meiga-project-v2 -n meiga-project-v2-ns
Release "meiga-release" has been upgraded. Happy Helming!
NAME: meiga-release
LAST DEPLOYED: Fri Nov  4 17:18:14 2022
NAMESPACE: meiga-project-v2-ns
STATUS: deployed
REVISION: 16
TEST SUITE: None
NOTES:
#templates/NOTEST.txt
***INSTRUCCIONES***
  Puedes acceder a la web Meiga-php en la dirección proporcionada por el ingress:
  $ kubectl describe ing meiga-release-ingress -n meiga-project-v2-ns
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```

4.4 Tests de unidad para Helm

Instalo el plugin de helm para hacer pruebas. Y nos explica como crear la char y la ayuda del comando.

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm plugin install https://github.com/quintush/helm-unittest
Support linux-amd64
Retrieving https://api.github.com/repos/quintush/helm-unittest/releases/latest
Downloading https://github.com/quintush/helm-unittest/releases/download/v0.2.9/helm-unittest-linux-amd64-0.2.9.tgz to location /tmp/_dist/
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 19.0M  100 19.0M    0     0  6013k      0  0:00:03  0:00:03 --:--:-- 8646k
Validating Checksum.
Preparing to install into /home/administrador/.local/share/helm/plugins/helm-unittest
helm-unittest installed into /home/administrador/.local/share/helm/plugins/helm-unittest
Running chart unittest written in YAML.

This renders your charts locally (without tiller) and
validates the rendered output with the tests defined in
test suite files. Simplest test suite file looks like
below:

---
# CHART_PATH/tests/deployment_test.yaml
suite: test my deployment
templates:
  - deployment.yaml
tests:
  - it: should be a Deployment
    asserts:
      - isKind:
          of: Deployment
---

Put the test files in "tests" directory under your chart
with suffix "_test.yaml", and run:

$ helm unittest my-chart

Or specify the suite files glob path pattern:

$ helm unittest -f 'my-tests/*.yaml' my-chart

Check https://github.com/quintush/helm-unittest for more
details about how to write tests.

Usage:
  unittest [flags] CHART [...]

Flags:
      --color                 enforce printing colored output even stdout is not a tty. Set to false to disable color
  -q, --failfast              direct quit testing, when a test is failed
  -f, --file stringArray      glob paths of test files location, default to tests/*_test.yaml (default [tests/*_test.yaml])
  -3, --helm3                 parse helm charts as helm3 charts
  -h, --help                  help for unittest
  -o, --output-file string    output-file the file where testresults are written in JUnit format, defaults no output is written to file
  -t, --output-type string    output-type the file-format where testresults are written in, accepted types are (JUnit, NUnit, XUnit) (default "
XUnit")
      --strict                strict parse the testsuites
  -u, --update-snapshot       update the snapshot cached if needed, make sure you review the change before update
  -v, --values stringArray    absolute or glob paths of values files location, default no values files
  -s, --with-subchart charts  include tests of the subcharts within charts folder (default true)
Installed plugin: unittest
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```

Copio los tests en la carpeta del proyecto. Queda este árbol de directorios y ficheros:

```
 % cp -r ../formacion/cursos/helm/codigo_practica_guiada_meiga/meiga-helm-v2/tests meiga-project-v2/.
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % tree meiga-project-v2
meiga-project-v2
├── charts
│   ├── ingress-nginx-4.3.0.tgz
│   └── mysql-9.4.1.tgz
├── Chart.yaml
├── my-values.yaml
├── templates
│   ├── configmap.yaml
│   ├── frontend-deploy.yaml
│   ├── frontend-service.yaml
│   ├── ingress.yaml
│   ├── NOTES.txt
│   └── secret.yaml
├── tests
│   ├── configmap_test.yaml
│   ├── frontend-deploy_test.yaml
│   ├── frontend-service_test.yaml
│   ├── ingress_test.yaml
│   └── secret_test.yaml
└── values.yaml

3 directories, 16 files
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```

Aplico las pruebas unitarias

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm unittest -3 meiga-project-v2

### Chart [ meiga-project ] meiga-project-v2

 PASS  test para configMap      meiga-project-v2/tests/configmap_test.yaml
 PASS  test para frontend-deploy        meiga-project-v2/tests/frontend-deploy_test.yaml
 PASS  test para frontend-service       meiga-project-v2/tests/frontend-service_test.yaml
 PASS  test para ingress        meiga-project-v2/tests/ingress_test.yaml
 PASS  test para secret meiga-project-v2/tests/secret_test.yaml

Charts:      1 passed, 1 total
Test Suites: 5 passed, 5 total
Tests:       6 passed, 6 total
Snapshot:    0 passed, 0 total
Time:        103.676867ms

administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```

5 Actividades

5.1 Actividade-1

Utilizaré el ingress creado en Kind con el script del curso de Kubernetes.

Primero voy a crear todos los artefactos en kubernetes sin helm para probar que funciona.

```
% mkdir wordpress
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
% cd wordpress
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wordpress
% touch bbdd-{deploy,service}.yaml configmap.yaml wp-{deploy,service}.yaml secret.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wordpress
% tree
.
├── bbdd-deploy.yaml
├── bbdd-service.yaml
├── configmap.yaml
├── secret.yaml
├── wp-deploy.yaml
└── wp-service.yaml

0 directories, 6 files
```

```
# configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: wp-config
  labels:
    tipo: "practica"
data:
  MYSQL_HOST: servicio-wp-bbdd
  MYSQL_USER: "root"
  MYSQL_DATABASE: "wp-db"


# bbdd-service.yaml
kind: Service
apiVersion: v1
metadata:
  name: servicio-wp-bbdd
```

```yaml
spec:
  selector:
    app: wp-mysql
  ports:  # esta é a parte de especificación propia
  - protocol: TCP
    port: 3306
    targetPort: 3306

# bbdd-deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-wp-bbdd
  labels:
    app: "wp-mysql"
spec:
  replicas: 1
  selector:
    matchLabels:
      app: wp-mysql
  template: # a partir de aquí definimos o pod
    metadata:
      labels:
        app: wp-mysql
    spec:
      containers:
      - name: wp-mysql
        image: mysql:8.0
        env:
          - name: "MYSQL_ROOT_PASSWORD"
            valueFrom:
              secretKeyRef:
```

```
            name: wp-secrets
            key: root-password
        ports:
        - containerPort: 3306

# secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: wp-secrets
type: Opaque
data:
  root-password: QzBudHI0c2VuNA==

# wp-service.yaml
kind: Service
apiVersion: v1
metadata:
  name: servicio-wp
spec:
  selector:
    app: deploy-wp
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80


# wp-deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
```

```
  name: deploy-wp
  labels:
    app: "deploy-wp"
    tipo: "practica"
spec:
  replicas: 1
  selector:
    matchLabels:
      app: deploy-wp
  template:
    metadata:
      labels:
        app: deploy-wp
    spec:
      containers:
      - name: deploy-wp
        image: wordpress:6.1-fpm-alpine
        env:
          - name: "MYSQL_PASSWORD"
            valueFrom:
              secretKeyRef:
                name: wp-secrets
                key: root-password
        envFrom:
          - configMapRef:
              name: wp-config
        ports:
        - containerPort: 80
```

Creo también el ingress.yaml:

```
# ingress.yaml
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
metadata:
  name: wp-ingress
  annotations:
      nginx.ingress.kubernetes.io/rewrite-target: "/"
      nginx.ingress.kubernetes.io/ssl-redirect: "false"
spec:
  rules:
  - http:
      paths:
      - pathType: Prefix
        path: /
        backend:
          service:
            name: servicio-wp
            port:
              number: 80
```

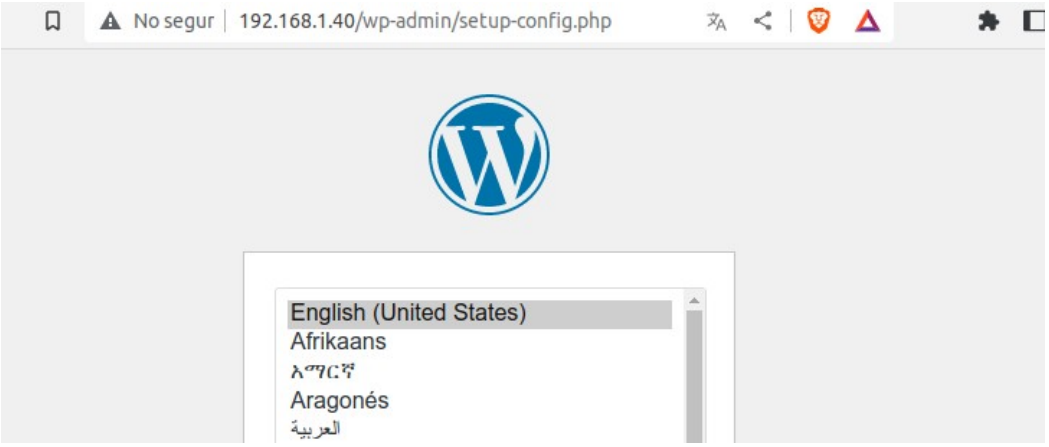
Creo un namespace:

```
% k create ns wp-proof
namespace/wp-proof created
```

Levanto toda la infraestructura:

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wordpress
 % k apply -f secret.yaml -n wp-proof
k apply -f bbdd-deploy.yaml -n wp-proof
k apply -f bbdd-service.yaml -n wp-proof
k apply -f wp-deploy.yaml -n wp-proof
k apply -f wp-service.yaml -n wp-proof
k apply -f ingress.yaml -n wp-proof
secret/wp-secrets created
deployment.apps/deploy-wp-bbdd created
service/servicio-wp-bbdd created
deployment.apps/deploy-wp created
service/servicio-wp created
ingress.networking.k8s.io/wp-ingress created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wordpress
 % k get ing -n wp-proof
NAME         CLASS    HOSTS   ADDRESS   PORTS   AGE
wp-ingress   <none>   *                 80      21s
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wordpress
 % k get all -n wp-proof
NAME                                  READY   STATUS    RESTARTS   AGE
pod/deploy-wp-6f86855ff-fhss9         1/1     Running   0          29s
pod/deploy-wp-bbdd-67cd54c76f-lc7xm   1/1     Running   0          29s

NAME                        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/servicio-wp         ClusterIP   10.96.219.35    <none>        80/TCP     29s
service/servicio-wp-bbdd    ClusterIP   10.96.248.112   <none>        3306/TCP   29s

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/deploy-wp         1/1     1            1           29s
deployment.apps/deploy-wp-bbdd    1/1     1            1           29s

NAME                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/deploy-wp-6f86855ff         1         1         1       29s
replicaset.apps/deploy-wp-bbdd-67cd54c76f   1         1         1       29s
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wordpress
 % k get ing -n wp-proof
NAME         CLASS    HOSTS   ADDRESS     PORTS   AGE
wp-ingress   <none>   *       localhost   80      34s
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wordpress
```

Compruebo:



Ahora vamos a crear la estructura helm en una carpeta

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm create wp-release
Creating wp-release
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % tree wp-release
wp-release
├── charts
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── hpa.yaml
│   ├── ingress.yaml
│   ├── NOTES.txt
│   ├── serviceaccount.yaml
│   ├── service.yaml
│   └── tests
│       └── test-connection.yaml
└── values.yaml

3 directories, 10 files
```

Borramos comentarios del fichero Chart.yaml para que se vea claro:

```
apiVersion: v2
name: wp-release
description: A Helm chart of Wordpress for Kubernetes
type: application
version: 0.1.0
appVersion: "1.16.0"
~
```

Borramos el contenido del fichero values.yaml

```
% echo "" > values.yaml
administrador@ubuntudocker ~/prueb
% cat values.yaml
```

Borramos todo lo que hay en la carpeta templates y copiamos los artefactos que nos han funcionado

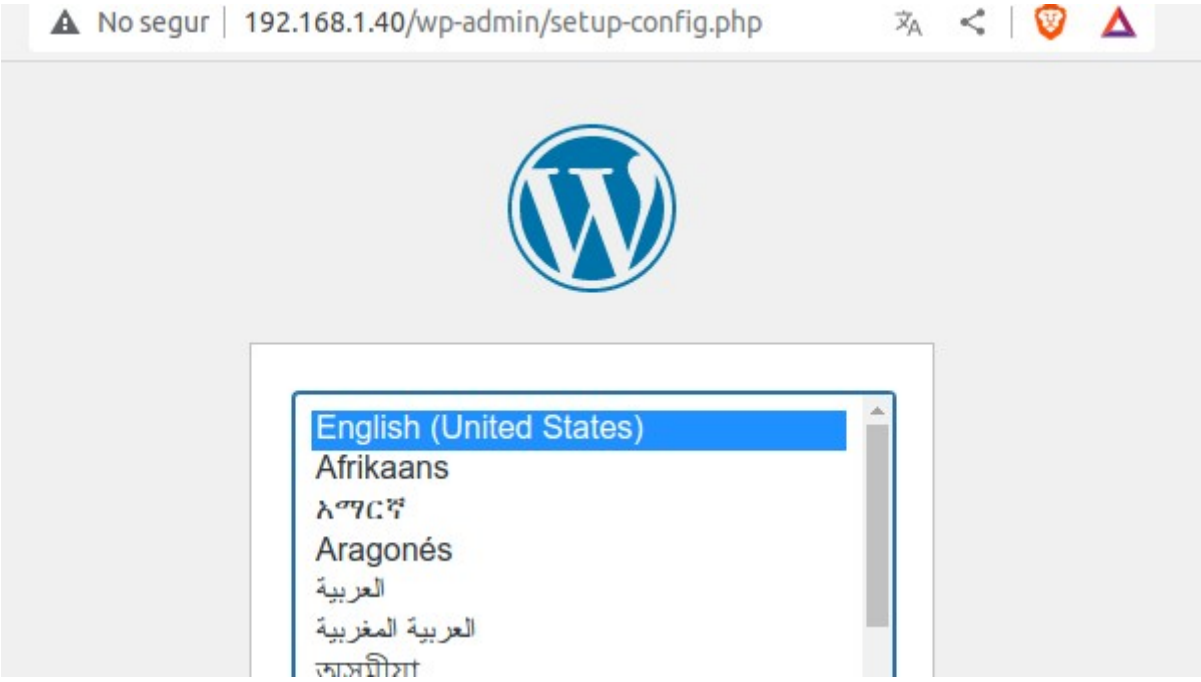
```
 % rm -rf templates/*
zsh: sure you want to delete all 8 files in /home/administrador/prueb
ates [yn]? y
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-release
 % ls templates
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-release
 % cp ../wordpress/* templates/.
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-release
 % tree
.
├── charts
├── Chart.yaml
├── templates
│   ├── bbdd-deploy.yaml
│   ├── bbdd-service.yaml
│   ├── configmap.yaml
│   ├── ingress.yaml
│   ├── secret.yaml
│   ├── wp-deploy.yaml
│   └── wp-service.yaml
└── values.yaml

2 directories, 9 files
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-release
```

Pruebo si la puedo lanzar con Helm, antes borro la prueba anterior:

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_he
 % k delete -f secret.yaml -n wp-proof
k delete -f bbdd-deploy.yaml -n wp-proof
k delete -f bbdd-service.yaml -n wp-proof
k delete -f wp-deploy.yaml -n wp-proof
k delete -f wp-service.yaml -n wp-proof
k delete -f ingress.yaml -n wp-proof
secret "wp-secrets" deleted
deployment.apps "deploy-wp-bbdd" deleted
service "servicio-wp-bbdd" deleted
deployment.apps "deploy-wp" deleted
service "servicio-wp" deleted
ingress.networking.k8s.io "wp-ingress" deleted
```

Creo otro namespaces e instalo con Helm:

```
administrador@ubuntudocker ~/pruebas_prefapp/mo
 % k create ns wp-release
namespace/wp-release created
administrador@ubuntudocker ~/pruebas_prefapp/mo
 % helm install wp-release . -n wp-release
NAME: wp-release
LAST DEPLOYED: Mon Nov  7 15:56:27 2022
NAMESPACE: wp-release
STATUS: deployed
REVISION: 1
TEST SUITE: None
administrador@ubuntudocker ~/pruebas_prefapp/mo
```



Antes de seguir adelante crearé volumenes para la persistencia tanto de la configuración wordpress como de la bbdd en otro release:

El volumen de wp:

```
# wp-volume.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-pv-claim
  labels:
    app: deploy-wp
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

En el manifiesto del deploy de wordpress de la release wp-release añado:

En containers de spec:

```
   volumeMounts:
        - name: wordpress-persistent-storage
          mountPath: /var/www/html
```

Y en spec:

```
   volumes:
      - name: wordpress-persistent-storage
        persistentVolumeClaim:
          claimName: wp-pv-claim
```

El volumen de la bbdd

```
#bbdd-volume.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wp-db-pv-claim
  labels:
    app: deploy-wp-bbdd
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

En el manifiesto del deploy de bbdd de la release wp-release añado:

En containers de spec:

```
        volumeMounts:
        - name: wp-db-persistent-storage
          mountPath: /var/lib/mysql
```

Y en spec:

```
      volumes:
      - name: wp-db-persistent-storage
        persistentVolumeClaim:
          claimName: wp-db-pv-claim
```

Compruebo que se despliegan los volumenes:

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-volume
 % k apply -f bbdd-volume.yaml -n wp-release && k apply -f wp-volume.yaml -n wp-
release
persistentvolumeclaim/wp-db-pv-claim created
persistentvolumeclaim/wp-pv-claim created
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-volume
```

Parece que todo ha ido bien

```
Context: kind-kind                    <0> all       <ctrl-d> Delete
Cluster: kind-kind                    <1> default   <d>      Describe
User:    kind-kind                                  <e>      Edit
K9s Rev: v0.26.7                                    <?>      Help
K8s Rev: v1.25.2                                    <u>      UsedBy
CPU:     n/a                                        <y>      YAML
MEM:     n/a

                    Persistentvolumeclaims(all)[2]
NAMESPACE↑  NAME            STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  AGE
wp-release  wp-db-pv-claim  Bound   pvc-bf1a292e-ae7c-4c28-9937-d0de3b80de02   5Gi       RWO           standard      37s
wp-release  wp-pv-claim     Bound   pvc-d0725fec-e04c-48c6-8b59-f6192902a175   5Gi       RWO           standard      37s
```

Elimino los volumenes, creo la release y coloco todo al sitio:

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-volume
 % helm create .
Creating .
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-volume
 % ls
bbdd-volume.yaml  charts  Chart.yaml  templates  values.yaml  wp-volume.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-volume
 % rm -rf templates/*
zsh: sure you want to delete all 8 files in /home/administrador/pruebas_prefapp/
modulo_helm/wp-volume/templates [yn]? y
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-volume
 % mv *-volume.yaml templates/.
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-volume
 % echo "" > values.yaml
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/wp-volume
 % tree
.
├── charts
├── Chart.yaml
├── templates
│   ├── bbdd-volume.yaml
│   └── wp-volume.yaml
└── values.yaml

2 directories, 4 files
```

Chart.yaml:

```
apiVersion: v2
name: wp-volume
description: A Helm chart of wp's volume for Kubernetes
type: application
version: 0.1.0
appVersion: "1.16.0"
~
~
```

Instalo la Chart de los volumenes:

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % helm install wp-volume wp-volume/ -n wp-release
NAME: wp-volume
LAST DEPLOYED: Wed Nov  9 09:17:06 2022
NAMESPACE: wp-release
STATUS: deployed
REVISION: 1
TEST SUITE: None
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
```

```
Ubuntu 20.04 practicas Kubernetes
Context: kind-kind                                  <0> all       <ctrl-d> Delete
Cluster: kind-kind                                  <1> default   <d>      Describe
User:    kind-kind                                                <e>      Edit
K9s Rev: v0.26.7                                                  <?>      Help
K8s Rev: v1.25.2                                                  <u>      UsedBy
CPU:     n/a                                                      <y>      YAML
MEM:     n/a
                                  Persistentvolumeclaims(all)[2]
NAMESPACE↑  NAME            STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  AGE
wp-release  wp-db-pv-claim  Bound   pvc-d06f84ac-b5e6-42b6-9319-e1e218c8ac71   5Gi       RWO           standard      53s
wp-release  wp-pv-claim     Bound   pvc-4603cdfe-7708-4587-82ce-36be23158fad   5Gi       RWO           standard      53s
```

Ahora voy a darle información en NOTES.txt de ambas Chart:

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm
 % cat wp-release/templates/NOTES.txt wp-volume/templates/NOTES.txt
############

La instalación de la release wp-release se compone en:
- Despliegue y servicio de BBDD
- Despliegue y servicio de WP
- Configmap
- Secrets
- Ingress

Es necesaria la instalación de la release wp-volume para darle persistenci
a a la bbdd mysql y a la configuración de wordpress.

############

############

La instalación de la release wp-volume se compone en:
- Volumen para bbdd mysql de la release "wp-release"
- Volumen para el CMS Wordpress de la release "wp-release"

Esta release no debería borrarse y en todo caso debe tener backups periódi
cos, ya que contiene los datos que deben ser persistentes de la release "w
p-release".

############
```

Compruebo que salen los mensajes con un upgrade

```
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/actividades-charts-helm
 % helm upgrade wp-volume wp-volume/ -n wp-release
Release "wp-volume" has been upgraded. Happy Helming!
NAME: wp-volume
LAST DEPLOYED: Wed Nov  9 09:34:18 2022
NAMESPACE: wp-release
STATUS: deployed
REVISION: 2
TEST SUITE: None
NOTES:
############

La instalación de la release wp-volume se compone en:
- Volumen para bbdd mysql de la release "wp-release"
- Volumen para el CMS Wordpress de la release "wp-release"

Esta release no debería borrarse y en todo caso debe tener backups periódicos, ya que cont
iene los datos que deben ser persistentes de la release "wp-release".

############
administrador@ubuntudocker ~/pruebas_prefapp/modulo_helm/actividades-charts-helm
 % helm upgrade wp-release wp-release/ -n wp-release
Release "wp-release" has been upgraded. Happy Helming!
NAME: wp-release
LAST DEPLOYED: Wed Nov  9 09:34:42 2022
NAMESPACE: wp-release
STATUS: deployed
REVISION: 2
TEST SUITE: None
NOTES:
############

La instalación de la release wp-release se compone en:
- Despliegue y servicio de BBDD
- Despliegue y servicio de WP
- Configmap
- Secrets
- Ingress

Es necesaria la instalación de la release wp-volume para darle persistencia a la bbdd mysq
l y a la configuración de wordpress.

############
```

La estructura queda así:

```
administrador@ubuntudocker ~/prueb
 % tree
.
├── wp-release
│   ├── charts
│   ├── Chart.yaml
│   ├── templates
│   │   ├── bbdd-deploy.yaml
│   │   ├── bbdd-service.yaml
│   │   ├── configmap.yaml
│   │   ├── ingress.yaml
│   │   ├── NOTES.txt
│   │   ├── secret.yaml
│   │   ├── wp-deploy.yaml
│   │   └── wp-service.yaml
│   └── values.yaml
└── wp-volume
    ├── charts
    ├── Chart.yaml
    ├── templates
    │   ├── bbdd-volume.yaml
    │   ├── NOTES.txt
    │   └── wp-volume.yaml
    └── values.yaml

6 directories, 15 files
```

Instalo el wordpress añadiendo la bbdd por la IU

Copio la estructura de directorios y cambio el ingress para que aparezca en una segunda release en /v2. (A la release 1 también le cambio el path a /v1)

```yaml
# ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: wp-ingress
  annotations:
      nginx.ingress.kubernetes.io/rewrite-target: "/"
      nginx.ingress.kubernetes.io/ssl-redirect: "false"
spec:
  rules:
  - http:
      paths:
      - pathType: Prefix
        path: /v2
        backend:
          service:
            name: servicio-wp
            port:
              number: 80
~
```

Creo un nuevo namespaces e instalo cambiando el nombre de la release. Actualizo la release1. Queda así:

release de helm

```
Context: kind-kind                    <0> all       <ctrl-d> Delete
Cluster: kind-kind                    <1> default   <d>      Describe
User:    kind-kind                                  <?>      Help
K9s Rev: v0.26.7                                    <v>      Values
K8s Rev: v1.25.2                                    <y>      YAML
CPU:     n/a
MEM:     n/a
                                        Helm(all)[4]
NAMESPACE↑      NAME            REVISION    STATUS      CHART               APP VERSION    AGE
wp-release1     wp-release1     8           deployed    wp-release-0.1.0    1.16.0         6m9s
wp-release1     wp-volume1      1           deployed    wp-volume-0.1.0     1.16.0         18m
wp-release2     wp-release2     3           deployed    wp-release-0.1.0    1.16.0         71s
wp-release2     wp-volume2      2           deployed    wp-volume-0.1.0     1.16.0         86s
```

Volumenes

```
Context: kind-kind                    <0> all       <ctrl-d> Delete
Cluster: kind-kind                    <1> default   <d>      Describe
User:    kind-kind                                  <e>      Edit
K9s Rev: v0.26.7                                    <?>      Help
K8s Rev: v1.25.2                                    <u>      UsedBy
CPU:     n/a                                        <y>      YAML
MEM:     n/a
                                        Persistentvolumeclaims(all)[4]
NAMESPACE↑      NAME            STATUS  VOLUME                                      CAPACITY  ACCESS MODES  STORAGECLASS  AGE
wp-release1     wp-db-pv-claim  Bound   pvc-fca34693-f869-469e-a1d2-912c1802265d    2Gi       RWO           standard      17m
wp-release1     wp-pv-claim     Bound   pvc-470ccf5b-cba2-4f72-a120-cb44191f45ca    2Gi       RWO           standard      17m
wp-release2     wp-db-pv-claim  Bound   pvc-a5a35fe3-b2f8-4c81-93cb-b9db215d3516    2Gi       RWO           standard      3m50s
wp-release2     wp-pv-claim     Bound   pvc-9bcbc84a-d3a8-42a7-8568-1ecd4c0b8604    2Gi       RWO           standard      3m50s
```

deploys

```
Context: kind-kind                    <0> all       <ctrl-d> Delete        <shift-f> Port-Fo…
Cluster: kind-kind                    <1> default   <d>      Describe      <r>       Restart
User:    kind-kind                                  <e>      Edit          <s>       Scale
K9s Rev: v0.26.7                                    <?>      Help          <y>       YAML
K8s Rev: v1.25.2                                    <l>      Logs
CPU:     n/a                                        <p>      Logs Previous
MEM:     n/a
                                        Deployments(all)[7]
NAMESPACE↑              NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
ingress-nginx           ingress-nginx-controller    1/1     1            1           2d2h
kube-system             coredns                     2/2     2            2           2d2h
local-path-storage      local-path-provisioner      1/1     1            1           2d2h
wp-release1             deploy-wp                   1/1     1            1           18m
wp-release1             deploy-wp-bbdd              1/1     1            1           18m
wp-release2             deploy-wp                   1/1     1            1           5m17s
wp-release2             deploy-wp-bbdd              1/1     1            1           5m17s
```

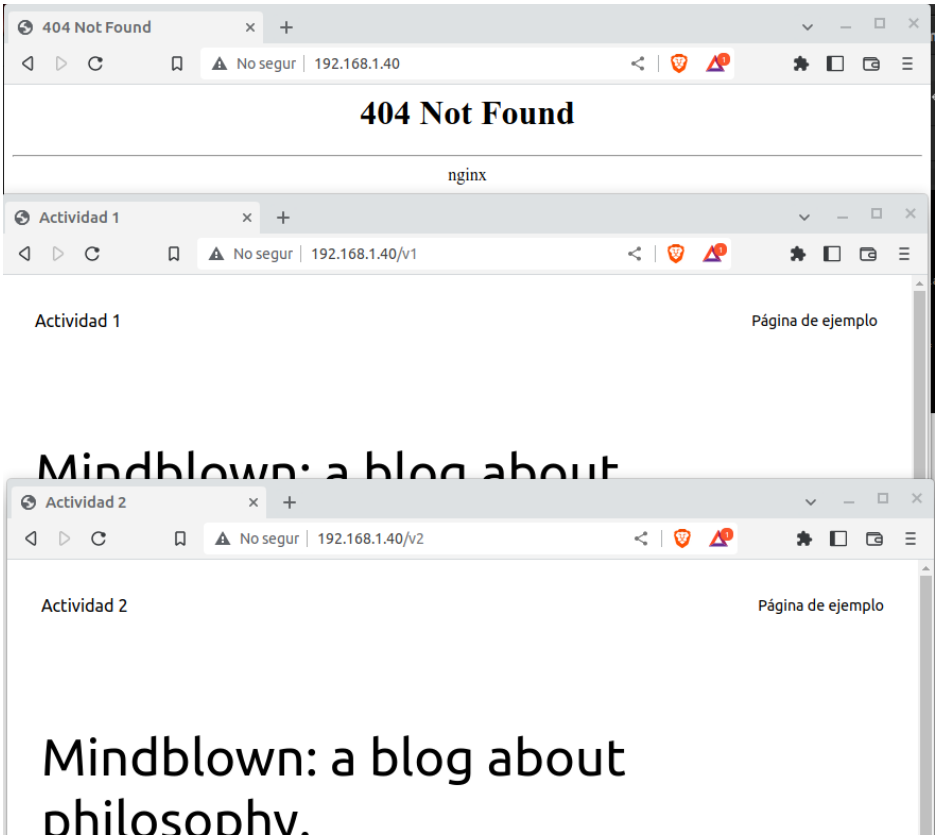
Pods

```
Context: kind-kind                          <0> all       <a>      Attach     <l>      Logs
Cluster: kind-kind                          <1> default   <ctrl-d> Delete     <p>      Logs Previous
User:    kind-kind                                        <d>      Describe   <shift-f> Port-Forward
K9s Rev: v0.26.7                                          <e>      Edit       <s>      Shell
K8s Rev: v1.25.2                                          <?>      Help       <n>      Show Node
CPU:     n/a                                              <ctrl-k> Kill       <f>      Show PortForw
MEM:     n/a

                                                   Pods(all)[20]
NAMESPACE↑          NAME                                          PF READY RESTARTS STATUS    IP           NODE           AGE
ingress-nginx       ingress-nginx-admission-create-fmm4q          ● 0/1          0 Completed 10.244.1.3   kind-worker    2d2h
ingress-nginx       ingress-nginx-admission-patch-9bqq6           ● 0/1          1 Completed 10.244.1.2   kind-worker    2d2h
ingress-nginx       ingress-nginx-controller-7d68cdddd8-n89vq     ● 1/1          2 Running   10.244.0.4   kind-control-plane 2d2h
kube-system         coredns-565d847f94-44psd                      ● 1/1          2 Running   10.244.0.5   kind-control-plane 2d2h
kube-system         coredns-565d847f94-g5ndm                      ● 1/1          2 Running   10.244.0.3   kind-control-plane 2d2h
kube-system         etcd-kind-control-plane                       ● 1/1          2 Running   172.20.0.4   kind-control-plane 2d2h
kube-system         kindnet-9pn4h                                 ● 1/1          2 Running   172.20.0.2   kind-worker2   2d2h
kube-system         kindnet-rcjkt                                 ● 1/1          2 Running   172.20.0.4   kind-control-plane 2d2h
kube-system         kindnet-x9l9k                                 ● 1/1          2 Running   172.20.0.3   kind-worker    2d2h
kube-system         kube-apiserver-kind-control-plane             ● 1/1          2 Running   172.20.0.4   kind-control-plane 2d2h
kube-system         kube-controller-manager-kind-control-plane    ● 1/1          2 Running   172.20.0.4   kind-control-plane 2d2h
kube-system         kube-proxy-9r6wb                              ● 1/1          2 Running   172.20.0.4   kind-control-plane 2d2h
kube-system         kube-proxy-l94bk                              ● 1/1          2 Running   172.20.0.2   kind-worker2   2d2h
kube-system         kube-proxy-tmdtz                              ● 1/1          2 Running   172.20.0.3   kind-worker    2d2h
kube-system         kube-scheduler-kind-control-plane             ● 1/1          2 Running   172.20.0.4   kind-control-plane 2d2h
local-path-storage  local-path-provisioner-684f458cdd-sd4cf       ● 1/1          4 Running   10.244.0.2   kind-control-plane 2d2h
wp-release1         deploy-wp-7db9c5bc7f-d9grp                    ● 1/1          0 Running   10.244.1.21  kind-worker    18m
wp-release1         deploy-wp-bbdd-67c95bc7bc-j5q7v               ● 1/1          0 Running   10.244.2.21  kind-worker2   18m
wp-release2         deploy-wp-7db9c5bc7f-ct8xp                    ● 1/1          0 Running   10.244.1.23  kind-worker    5m42
wp-release2         deploy-wp-bbdd-67c95bc7bc-xphvd               ● 1/1          0 Running   10.244.2.23  kind-worker2   5m42
```

ingress

```
Context: kind-kind                          <0> all       <ctrl-d> Delete
Cluster: kind-kind                          <1> default   <d>      Describe
User:    kind-kind                                        <e>      Edit
K9s Rev: v0.26.7                                          <?>      Help
K8s Rev: v1.25.2                                          <y>      YAML
CPU:     n/a
MEM:     n/a

                                                   Ingresses(all)[2]
NAMESPACE↑          NAME          CLASS          HOSTS          ADDRESS          PORTS          AGE
wp-release1         wp-ingress    <none>         *              localhost        80             20m
wp-release2         wp-ingress    <none>         *              localhost        80             7m28s
```

Ahora no hay nada en la raíz, están en la URI v1 y v2



5.2 Actividade-2

Incluyo en los releases el .helmignore de prefapp-helm:

```
# Patterns to ignore when building packages.
# This supports shell glob matching, relative path matching, and
# negation (prefixed with !). Only one pattern per line.
.DS_Store
# Common VCS dirs
.git/
.gitignore
.bzr/
.bzrignore
.hg/
.hgignore
.svn/
# Common backup files
*.swp
*.bak
*.tmp
*.orig
*~
# Various IDEs
.project
.idea/
*.tmproj
.vscode/


# Según prefapp-helm ignorar también
#


templates/_renders_cert_issuer.yaml
templates/_stash.yaml

tests/

README.md

*.sh

docs/

a.yml

*.tgz

oldtests/
~
~
```

Hago los cambios de cada uno de los artefactos.

Lo dejo todo en https://github.com/manuelver/formacion/tree/feature/helm-exerc-mvergara/cursos/helm/00_solucions/99_other_wp