

Seminario DevOps

Práctica formativa N°3

Kubernetes

Noviembre 2025

Abstract:

Este informe describe la instalación y uso de Minikube y Kubernetes, junto con el despliegue de una aplicación mediante contenedores, deployments y services.

RACHICH, Alejandra Lorena

Comisión 3D



Contenidos

Práctica Formativa Obligatoria N° 3	1
Ejecución de la práctica (Parte I).....	1
Paso 1: Instalación y arranque de Minikube.....	1
Paso 2: kubectl get po -A	1
Paso 3: minikube dashboard.....	2
Paso 4: Creación de un deployment de prueba.....	3
Paso 5: kubectl port-forward	5
Paso 6: Addons.....	5
Ejecución de la práctica (Parte II).....	7
Creación de archivos y explicación	7
Link al repositorio en GitHub:	7
Construir imagen Docker	7
Comprobar imagen	8
Generar y describir YAML	8
Aplicar YAML y resolver error	9
Crear servicio para la app PHP	10
Acceder al servicio	10
Problemas y soluciones experimentados	11
Problema 1: PROVIDER_VIRTUALBOX_NOT_FOUND	11
Problema 2: HOST_VIRT_UNAVAILABLE (Conflicto de virtualización).....	12
Problema 3: imagepullbackoff en la parte II.....	12



Práctica Formativa Obligatoria N° 3

Ejecución de la práctica (Parte I)

A continuación, se detallan los comandos ejecutados y su explicación, siguiendo los pasos de la práctica.

Paso 1: Instalación y arranque de Minikube

- **Comando:** minikube start --driver=virtualbox
- **Explicación:** Inicia el clúster de Kubernetes usando Minikube, especificando que debe usar virtualbox como *driver* para crear la máquina virtual.

```
C:\Users\Lore>minikube start --driver=virtualbox
* minikube v1.37.0 on Microsoft Windows 10 Home 10.0.19045.6456 Build 19045.6456
E1029 10:45:48.045033    5508 start.go:829] api.Load failed for minikube: filestore "minikube": Docker machine "minikube" does not exist. Use "docker-machine ls" to list machines. Use "docker-machine create" to add a new one.
E1029 10:45:48.096936    5508 start.go:829] api.Load failed for minikube: filestore "minikube": Docker machine "minikube" does not exist. Use "docker-machine ls" to list machines. Use "docker-machine create" to add a new one.
* Using the virtualbox driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Creating virtualbox VM (CPUs=2, Memory=4000MB, Disk=20000MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube VM
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
* Configuring bridge CNI (Container Networking Interface) ...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Paso 2: kubectl get po -A

- **Comando:** kubectl get po -A
- **Explicación:** Este comando solicita (get) todos los **Pods** (po) que se encuentran en **todos los namespaces** (-A o --all-namespaces). La salida muestra los componentes internos de Kubernetes que corren en el namespace kube-system (como coredns, etcd, kube-apiserver, etc.), que son el "cerebro" del clúster.

```
C:\Users\Lore>kubectl get po -A
NAMESPACE      NAME                           READY   STATUS    RESTARTS   AGE
kube-system    coredns-66bc5c9577-15bdg     1/1     Running   0          15m
kube-system    etcd-minikube                 1/1     Running   0          15m
kube-system    kube-apiserver-minikube       1/1     Running   0          15m
kube-system    kube-controller-manager-minikube 1/1     Running   0          15m
kube-system    kube-proxy-6ccz8                1/1     Running   0          15m
kube-system    kube-scheduler-minikube        1/1     Running   0          15m
kube-system    storage-provisioner           1/1     Running   0          15m
```



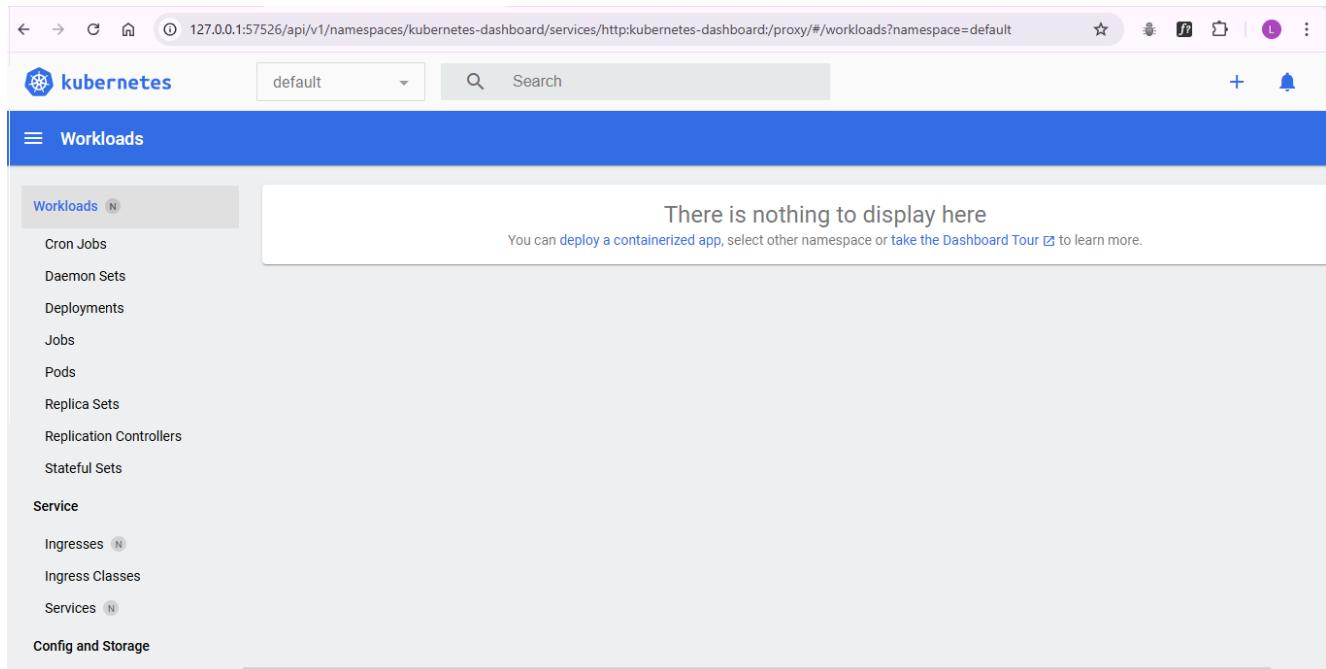
Paso 3: minikube dashboard

- **Comando:** minikube dashboard
- **Explicación:** Es un comando de Minikube que activa el *addon* del Dashboard (si no lo está) y abre la interfaz gráfica web oficial de Kubernetes en el navegador. Esta interfaz permite gestionar y visualizar el clúster (Deployments, Pods, Services) de forma gráfica.
- **Explicación (Cargas de Trabajo):** El dashboard muestra varias "Cargas de trabajo" (Workloads):
 - **Deployments:** Define el estado deseado de una aplicación (ej: "3 copias de mi app"). Gestiona actualizaciones y *rollbacks*.
 - **Pods:** La unidad más pequeña del clúster.
 - **Replicsets:** Asegura que un número específico de Pods esté siempre corriendo. Es usado por los deployments.
 - **Statefulsets:** Similar a los deployments, pero para aplicaciones "con estado" (como bases de datos) que necesitan almacenamiento persistente e ids fijos.
 - **Jobs / cronjobs:** Para tareas que se ejecutan una vez y terminan (Jobs) o que se ejecutan de forma programada (cronjobs).

```
C:\Users\Lore>minikube dashboard
* Enabling dashboard ...
- Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
- Using image docker.io/kubernetesui/dashboard:v2.7.0
* Some dashboard features require the metrics-server addon. To enable all features please run:
  minikube addons enable metrics-server

* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:57526/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```





The screenshot shows the Kubernetes Dashboard interface. At the top, there's a header bar with icons for back, forward, refresh, and search, along with a URL field showing the connection to the local Minikube cluster. Below the header is a navigation bar with tabs: 'Workloads' (selected), 'Ingresses', 'Services', and 'Config and Storage'. On the left, a sidebar lists various resource types: Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, Ingress Classes, and Services. The main content area is titled 'Workloads' and displays a message: 'There is nothing to display here. You can deploy a containerized app, select other namespace or take the Dashboard Tour to learn more.' A small blue button at the bottom right of the content area says '+ Deploy'.

Paso 4: Creación de un deployment de prueba

- **Comando 4a (Crear Deployment):** kubectl create deployment hello-world-minikube --image=kicbase/echo-server:1.0
 - **Explicación:** Crea un nuevo objeto Deployment llamado hello-world-minikube usando una imagen de prueba (echo-server) de internet.

```
C:\Users\Lore>kubectl create deployment hello-world-minikube --image=kicbase/echo-server:1.0
deployment.apps/hello-world-minikube created
```



Workload Status

Deployments Pods Replica Sets

Deployments

Name	Images	Labels	Pods	Created ↑
hello-world-minikube	kicbase/echo-server:1.0	app: hello-world-minikube	1 / 1	16 minutes ago

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
hello-world-minikube-745f69589d-w828b	kicbase/echo-server:1.0	app: hello-world-minikube pod-template-hash: 745f69589d	minikube	Running	0	-	-	16 minutes ago

Replica Sets

Name	Images	Labels	Pods	Created ↑
hello-world-minikube-745f69589d	kicbase/echo-server:1.0	app: hello-world-minikube pod-template-hash: 745f69589d	1 / 1	16 minutes ago

- **Comando 4b (Exponer servicio):** kubectl expose deployment hello-world-minikube --type=nodeport --port=8080
 - **Explicación:** Crea un "puente" (un objeto Service) para acceder al Deployment.
- **Explicación (Parámetros -type y -port):**
 - **--type=nodeport:** Indica que el servicio debe ser accesible desde fuera del clúster, abriendo un puerto en la IP de la máquina virtual (Nodo) de Minikube.
 - **--port=8080:** Especifica el puerto en el que la aplicación está escuchando dentro del Pod. El Service redirigirá el tráfico a este puerto.

```
C:\Users\Lore>kubectl expose deployment hello-world-minikube --type=NodePort --port=8080
service/hello-world-minikube exposed
```

- **Comando 4c (Acceder al servicio):** minikube service hello-world-minikube
 - **Explicación:** Es un comando de ayuda que obtiene automáticamente la URL (IP y nodeport) del servicio y la abre en el navegador.
 - **Resultado:** El navegador mostró la respuesta "echo" del Pod, confirmando el éxito.



```
C:\Users\Lore>minikube service hello-world-minikube
NAMESPACE | NAME | TARGET PORT | URL
default   | hello-world-minikube | 8080 | http://192.168.59.100:32100
* Opening service default/hello-world-minikube in default browser...
```

```
Não seguro 192.168.59.100:32100
Request served by hello-world-minikube-745f69589d-w828b
HTTP/1.1 GET /
Host: 192.168.59.100:32100
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,pt-BR;q=0.8,pt;q=0.7,es-AR;q=0.6,es;q=0.5
Cache-Control: max-age=0
Connection: keep-alive
Dnt: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36
```

Paso 5: kubectl port-forward

- **Comando:** kubectl port-forward service/hello-world-minikube 7080:8080
- **Explicación:** Es un método alternativo para acceder a un servicio, usado principalmente para *debug*. Crea un túnel de red directo desde la máquina local (localhost:7080) al puerto del servicio (8080), sin necesidad de un nodeport. Solo funciona mientras el comando se está ejecutando.

```
C:\Users\Lore>kubectl port-forward service/hello-world-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
Handling connection for 7080
Handling connection for 7080
```

```
localhost:7080
Request served by hello-world-minikube-745f69589d-w828b
HTTP/1.1 GET /
Host: localhost:7080
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9,pt-BR;q=0.8,pt;q=0.7,es-AR;q=0.6,es;q=0.5
Connection: keep-alive
Dnt: 1
Sec-Ch-Ua: "Google Chrome";v="141", "Not?A_Brand";v="8", "Chromium";v="141"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36
```

Paso 6: Addons

- **Comando:** minikube addons list
- **Explicación (3 addons más usados):**
 1. **Dashboard (enabled):** La interfaz gráfica web que ya estabamos usando.



2. **Ingress (disabled):** Un controlador avanzado de tráfico. Permite gestionar el acceso externo a múltiples servicios usando reglas (ej. Miweb.com/app1, miweb.com/app2), todo por el puerto 80/443. Es la forma estándar de exponer servicios en producción.
3. **Metrics-server (disabled):** Recolecta métricas de uso de CPU y Memoria de los Pods. Es esencial para el monitoreo y para que el dashboard muestre gráficos de consumo.

ADDON NAME	PROFILE	STATUS	MAINTAINER
ambassador	minikube	disabled	3rd party (Ambassador)
amd-gpu-device-plugin	minikube	disabled	3rd party (AMD)
auto-pause	minikube	disabled	minikube
cloud-spanner	minikube	disabled	Google
csi-hostpath-driver	minikube	disabled	Kubernetes
dashboard	minikube	enabled	Kubernetes
default-storageclass	minikube	enabled	Kubernetes
efk	minikube	disabled	3rd party (Elastic)
freshpod	minikube	disabled	Google
gcp-auth	minikube	disabled	Google
gvisor	minikube	disabled	minikube
headlamp	minikube	disabled	3rd party (kinvolk.io)
inaccel	minikube	disabled	3rd party (InAccel [info@inaccel.com])
ingress	minikube	disabled	Kubernetes
ingress-dns	minikube	disabled	minikube
inspektor-gadget	minikube	disabled	3rd party (inspektor-gadget.io)
istio	minikube	disabled	3rd party (Istio)
istio-provisioner	minikube	disabled	3rd party (Istio)
kong	minikube	disabled	3rd party (Kong HQ)
kubeflow	minikube	disabled	3rd party
kubetail	minikube	disabled	3rd party (kubetail.com)
kubevirt	minikube	disabled	3rd party (KubeVirt)
logviewer	minikube	disabled	3rd party (unknown)
metallb	minikube	disabled	3rd party (MetallLB)
metrics-server	minikube	disabled	Kubernetes
nvidia-device-plugin	minikube	disabled	3rd party (NVIDIA)
nvidia-driver-installer	minikube	disabled	3rd party (NVIDIA)
nvidia-gpu-device-plugin	minikube	disabled	3rd party (NVIDIA)
olm	minikube	disabled	3rd party (Operator Framework)
pod-security-policy	minikube	disabled	3rd party (unknown)
portainer	minikube	disabled	3rd party (Portainer.io)
registry	minikube	disabled	minikube
registry-aliases	minikube	disabled	3rd party (unknown)
registry-creds	minikube	disabled	3rd party (UPMC Enterprises)
storage-provisioner	minikube	enabled	minikube
storage-provisioner-gluster	minikube	disabled	3rd party (Gluster)
storage-provisioner-rancher	minikube	disabled	3rd party (Rancher)
volcano	minikube	disabled	third-party (volcano)
volumesnapshots	minikube	disabled	Kubernetes
yakd	minikube	disabled	3rd party (marcnuri.com)



Ejecución de la práctica (Parte II)

Creación de archivos y explicación

- Acción:** Se creó un repositorio DevOps-PFO-3 con un Dockerfile y una subcarpeta src/index.php.

Link al repositorio en GitHub:

https://github.com/Alexa-2k/DevOps-PFO_3-Kubernetes

- Explicación (/var/www/html y chown):**

- **/var/www/html:** Es el directorio raíz estándar para servidores web (como Apache o Nginx). La imagen base de PHP está configurada para servir archivos desde allí.
- **--chown=nobody:** Es una medida de seguridad. Cambia el propietario de los archivos a un usuario sin privilegios (nobody), previniendo que un atacante obtenga control root si explota una vulnerabilidad en la app.

Construir imagen Docker

- Comando (Conexión):** & minikube -p minikube docker-env | Invoke-Expression
 - **Explicación:** Este comando conecta la terminal local al *daemon* de Docker que corre *dentro* de la máquina virtual de Minikube. Esto asegura que la imagen se construya en el lugar correcto.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> minikube start --driver=virtualbox
* minikube v1.37.0 on Microsoft Windows 10 Home 10.0.19045.6456 Build 19045.6456
* Using the virtualbox driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Restarting existing virtualbox VM for "minikube" ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube VM
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
* Configuring bridge CNI (Container Networking Interface) ...
- Using image docker.io/kubernetessui/dashboard:v2.7.0
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
- Using image docker.io/kubernetessui/metrics-scraper:v1.0.8
* Verifying Kubernetes components...
* Some dashboard features require the metrics-server addon. To enable all features please run:

    minikube addons enable metrics-server

* Enabled addons: storage-provisioner, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\WINDOWS\system32> & minikube -p minikube docker-env | Invoke-Expression
PS C:\WINDOWS\system32> cd \
PS C:\> cd Users\lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3
```

- Comando (Build):** docker build -t mi-app-php .

- **Explicación:** Construye la imagen de Docker usando el Dockerfile en la carpeta actual (.) Y le asigna el nombre (-t) mi-app-php.



```

PS C:\WINDOWS\system32> cd \
PS C:> cd Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3
[+] Building 10.7s (10/10) FINISHED
      docker:default
-> [internal] load build definition from Dockerfile          0.2s
=> => transferring dockerfile: 221B                         0.2s
-> [internal] load metadata for docker.io/library/php:fpm-alpine   5.1s
-> [auth] library/php:pull token for registry-1.docker.io    0.0s
-> [internal] load .dockerrignore                           0.0s
-> => transferring context: 28                            0.0s
-> [1/4] FROM docker.io/library/php:fpm-alpine@sha256:994527feb1fe228e1f28f657abb6ae0e835b68b2dc0c 4.4s
-> => resolve docker.io/library/php:fpm-alpine@sha256:994527feb1fe228e1f28f657abb6ae0e835b68b2dc0c 0.0s
-> => sha256:994527feb1fe228e1f28f657abb6ae0e835b68b2dc0c8ef3d05c3fa06416371a 10.31kB / 10.31kB  0.0s
-> => sha256:47bf062eed91850e9eef639f7785d294e0217ff17d118cc2b508893a9e13dbbba 933B / 933B  0.7s
-> => sha256:fa23c6dc94d5e58120c3c06958c897b78601252ff44f0b50f3a59ba78ba4594 9.88kB / 9.88kB  0.0s
-> => sha256:b9761938e7b22d04283840c8554b214e8d806297cc0c17aedf5285070071de32 3.25kB / 3.25kB  0.0s
-> => sha256:2d35ebdb57d9971fea0cac1582aa78935adf8058b2cc32db163c98822e5dfab 3.80MB / 3.80MB  0.5s
-> => sha256:7b16f419f0e84d206d1c75bb7e7dc82df08577c70c1a4666c51198d49e473155 3.46MB / 3.46MB  0.7s
-> => extracting sha256:2d35ebdb57d9971fea0cac1582aa78935adf8058b2cc32db163c98822e5dfab 0.4s
-> => sha256:bb3739b494ab73f74536975c1d0786c7432eaacec564f8e1c70d414795320 222B / 222B  0.9s
-> => sha256:9f10ba94659a998829a6d7a9ca43c3230072facb64e82e187c031a6935bbaa89 13.67MB / 13.67MB  1.8s
-> => sha256:f0ded0a5d3cf08eb3bbb2ebbe2e2c31bcd6a8ff9ad3ae6f6ca214a678e4dc 484B / 484B  1.2s
-> => sha256:783bb9d009dbe9189b962849a49d19d820a1e97a17b2d3a5050917c481f2c0 15.03MB / 15.03MB  2.3s
-> => extracting sha256:7b16f419f0e84d206d1c75bb7e7dc82df08577c70c1a4666c51198d49e473155 0.8s
-> => sha256:0212706b5f97db8f7bdde5809473b6a967163920afffd25ef5b5a082fb734a3a 2.45kB / 2.45kB  1.7s
-> => sha256:731fc6ecbbe00d75552781db246d614759e94c437791236dad9f5cdc579d80d5 20.08kB / 20.08kB  2.1s
-> => sha256:78761c868beb8f46ff256cc8a2cad5d4dbc2810e8b39b3b002ee7e6dd3763b260 20.07kB / 20.07kB  2.2s
-> => extracting sha256:47bf062eed91850e9eef639f785d294e0217ff17d118cc2b508893a9e13dbbba 0.0s
-> => extracting sha256:bb3739b494ab73f745d356975c1d0786c7432eaacec564f8e1c70d414795320 0.0s
-> => sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1 32B / 32B  2.5s
-> => extracting sha256:9f10ba94659a998829a6d7a9ca43c3230072facb64e82e187c031a6935bbaa89 0.2s
-> => sha256:a5fb1c507fe27657440005412bfdcd17a8524ea7273820a3cb167a4440e64d 9.19kB / 9.19kB  2.6s
-> => extracting sha256:f0ded0a5d3cf08eb3bbb2ebbe2e2c31bcd6a8ff9ad3ae6f6ca214a678e4dc 0.0s
-> => extracting sha256:783bb9d009dbe9189b962849a49d19d820a1e4c97a17b2d3a5050917c481f2c0 1.6s
-> => extracting sha256:0212706b5f97db8f7bdde5809473b6a967163920afffd25ef5b5a082fb734a3a 0.0s
-> => extracting sha256:731fc6ecbbe00d75552781db246d614759e94c437791236dad9f5cdc579d80d5 0.0s
-> => extracting sha256:78761c868beb8f46ff256cc8a2cad5d4dbc2810e8b39b3b002ee7e6dd3763b260 0.0s
-> => extracting sha256:4f4fb700ef54461cfa02571ae0db9a0dc1e0cdb5577484a6d75e68dc38e8acc1 0.0s
-> => extracting sha256:a5fb1c507fe27657440005412bfdcd17a8524ea7273820a3cba167a4440e64d 0.0s
-> [internal] load build context                           0.0s
-> => transferring context: 1478                          0.0s
-> [2/4] RUN mkdir -p /var/www/html                      0.4s
-> [3/4] COPY --chown=nobody src/ /var/www/html/         0.0s
-> [4/4] WORKDIR /var/www/html                          0.0s
-> exporting to image                                    0.0s
-> => exporting layers                                 0.0s
-> => writing image sha256:668f51a0ca536f155db537b36c8087a62d29b1f148f8257debd356a7d8ca4bb8 0.0s
-> => naming to docker.io/library/mi-app-php           0.0s
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3>

```

Comprobar imagen

- Comando:** minikube image ls
- Explicación:** Lista las imágenes de Docker disponibles dentro de Minikube. Se usó para confirmar que la imagen mi-app-php:latest se había construido correctamente.

```

PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> minikube image ls
registry.k8s.io/pause:3.10.1
registry.k8s.io/kube-scheduler:v1.34.0
registry.k8s.io/kube-proxy:v1.34.0
registry.k8s.io/kube-controller-manager:v1.34.0
registry.k8s.io/kube-apiserver:v1.34.0
registry.k8s.io/etcld:3.6.4-0
registry.k8s.io/coredns/coredns:v1.12.1
gcr.io/k8s-minikube/storage-provisioner:v5
docker.io/library/mi-app-php:latest
docker.io/kubernetesui/metrics-scrapers:<none>
docker.io/kubernetcntr/dashboard:<none>
docker.io/kicbase/echo-server:1.0
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3>

```

Generar y describir YAML

- Comando:** kubectl create deployment mi-app-php-deployment --image=mi-app-php --dry-run=client -o yaml > deploy-mi-app-php.yaml



- **Explicación:** Este comando no crea el *deployment* directamente.
 - `--dry-run=client`: Simula la creación localmente.
 - `-o yaml`: Indica que la salida debe ser en formato YAML.
 - `> deploy...:` Redirige la salida a un archivo.
- **Descripción:** El archivo YAML es una "plantilla" declarativa que define el estado deseado del *deployment* (imagen, nombre, réplicas, etc.). Es la forma preferida de gestionar recursos en Kubernetes.

```
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl create deployment mi-app-php-deployment --image=mi-app-php --dry-run=client -o yaml > deploy-mi-app-php.yaml
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> ls

Directory: C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3

Mode                LastWriteTime         Length Name
----                -----          ----
d-----        10/31/2025 11:11 PM            capturas
d-----        10/31/2025 10:48 PM            src
-a----        10/31/2025 11:13 PM        850 deploy-mi-app-php.yaml
-a----        10/31/2025 10:48 PM        182 Dockerfile
```

Aplicar YAML y resolver error

- **Comando:** `kubectl apply -f deploy-mi-app-php.yaml`
- **Acción:** Este comando lee el archivo YAML y le ordena a Kubernetes crear o actualizar los recursos para que coincidan con la configuración.
- **Solución:** Como se mencionó en la Sección 1, el primer intento falló con `imagepullbackoff`. Se editó el YAML para agregar `imagepullpolicy: Never` y se volvió a ejecutar el comando `kubectl apply`.

```
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl apply -f deploy-mi-app-php.yaml
deployment.apps/mi-app-php-deployment created
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> ■
```

```
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl apply -f deploy-mi-app-php.yaml
deployment.apps/mi-app-php-deployment created
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl get deployments
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
hello-world-minikube   1/1     1          1          2d12h
mi-app-php-deployment 0/1     1          0          105s
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> ■
```

- **Comando (Verificación):** `kubectl get pods`
- **Resultado:** El nuevo pod `mi-app-php-deployment...` Se mostró con estado `Running`.

```
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
hello-world-minikube-745f69589d-w828b   1/1     Running   1 (35m ago)   2d12h
mi-app-php-deployment-7fb6bc7bbc-54wrv   0/1     ImagePullBackOff   0          2m42s
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3>
```



```
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
hello-world-minikube-745f69589d-w828b  1/1     Running   1 (35m ago)  2d12h
mi-app-php-deployment-7fb6bc7bbc-54wrwv  0/1     ImagePullBackOff  0          2m42s
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl apply -f deploy-mi-app-php.yaml
deployment.apps/mi-app-php-deployment configured
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
hello-world-minikube-745f69589d-w828b  1/1     Running   1 (39m ago)  2d12h
mi-app-php-deployment-f4bf994fb-xgwkx  1/1     Running   0          28s
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3>
```

Crear servicio para la app PHP

- Comando (Expo):** kubectl expose deployment/mi-app-php-deployment --type="nodeport" --port 80
- Comando (Verificación):** kubectl get services
- Explicación:** Se expuso el *deployment* de PHP en el puerto 80 (definido en el Dockerfile) usando nodeport.

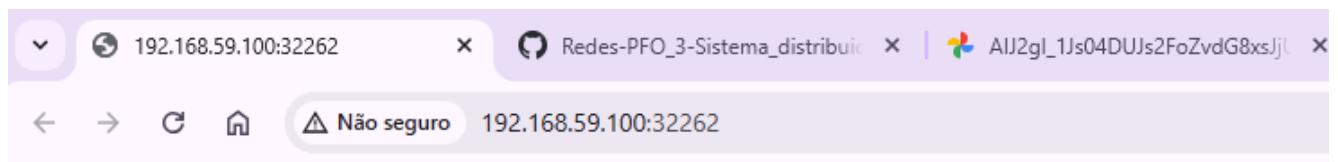
```
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl expose deployment/mi-app-php-deployment --type="NodePort" --port 80
service/mi-app-php-deployment exposed
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> kubectl get services
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
hello-world-minikube   NodePort   10.107.229.16 <none>        8080:32100/TCP  2d11h
kubernetes       ClusterIP  10.96.0.1    <none>        443/TCP       2d12h
mi-app-php-deployment  NodePort   10.104.251.204 <none>        80:32262/TCP   13s
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3>
```

Acceder al servicio

- Comando:** minikube service mi-app-php-deployment
- Explicación:** Se usó el comando de ayuda de Minikube para abrir automáticamente la URL del nuevo servicio en el navegador.
- Resultado:** El navegador mostró el mensaje del archivo index.php: Seminario devops Este es mi repositorio... Seguido del *hostname* del Pod, confirmando que la aplicación PHP personalizada se estaba ejecutando correctamente dentro de Kubernetes.

```
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3> minikube service mi-app-php-deployment
NAME         NAMESPACE   TARGET PORT   URL
mi-app-php-deployment default     80          http://192.168.59.100:32262
* Opening service default/mi-app-php-deployment in default browser...
PS C:\Users\Lore\Documents\00programacion\01.TDS\05_cuatr\Seminario-DevOps\DevOps-PFO-3>
```





Seminario DevOps Este es mi repositorio mi-app-php-deployment-f4bf994fb-xgwkx

Problemas y soluciones experimentados

Durante la instalación y configuración inicial de Minikube, surgieron varios inconvenientes. Esta sección detalla los problemas encontrados y las soluciones aplicadas.

Problema 1: PROVIDER_VIRTUALBOX_NOT_FOUND

- **Comando ejecutado:** minikube start --driver=virtualbox
- **Error:** La terminal devolvió el error PROVIDER_VIRTUALBOX_NOT_FOUND, indicando que no podía encontrar el *driver* de virtualbox.
- **Solución:** Aunque Minikube estaba instalado, el *hypervisor* (virtualbox) no lo estaba. La solución fue descargar e instalar virtualbox desde su página web oficial.

```
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lore>minikube start --driver=virtualbox
* minikube v1.37.0 on Microsoft Windows 10 Home 10.0.19045.6456 Build 19045.6456
* Using the virtualbox driver based on user configuration

* Exiting due to PROVIDER_VIRTUALBOX_NOT_FOUND: The 'virtualbox' provider was not found: unable to find VBoxManage in $PATH
* Suggestion: Install VirtualBox
* Documentation: https://minikube.sigs.k8s.io/docs/reference/drivers/virtualbox/
```

```
Microsoft Windows [Version 10.0.19045.6456]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lore>minikube start --driver=virtualbox
* minikube v1.37.0 on Microsoft Windows 10 Home 10.0.19045.6456 Build 19045.6456
* Using the virtualbox driver based on user configuration

* Exiting due to PROVIDER_VIRTUALBOX_NOT_FOUND: The 'virtualbox' provider was not found: unable to find VBoxManage in $PATH
* Suggestion: Install VirtualBox
* Documentation: https://minikube.sigs.k8s.io/docs/reference/drivers/virtualbox/

C:\Users\Lore>minikube start --driver=virtualbox
* minikube v1.37.0 on Microsoft Windows 10 Home 10.0.19045.6456 Build 19045.6456
* Using the virtualbox driver based on user configuration

* Exiting due to PROVIDER_VIRTUALBOX_NOT_FOUND: The 'virtualbox' provider was not found: unable to find VBoxManage in $PATH
* Suggestion: Install VirtualBox
* Documentation: https://minikube.sigs.k8s.io/docs/reference/drivers/virtualbox/

C:\Users\Lore>
```



Problema 2: HOST_VIRT_UNAVAILABLE (Conflicto de virtualización)

- **Comando ejecutado:** minikube start --driver=virtualbox (después de instalar virtualbox).
- **Error:** La terminal devolvió This computer doesn't have VT-X/AMD-v enabled. Enabling it in the BIOS is mandatory.
- **Solución:**
 1. Se verificó la **BIOS/UEFI** del equipo y se confirmó que la Virtualization Technology **ya estaba habilitada**.
 2. El problema era un conflicto a nivel de sistema operativo: **Hyper-V de Windows** estaba "secuestrando" la virtualización, impidiendo que virtualbox la usara.
 3. La solución fue ir a "Activar o desactivar las características de Windows" y **desmarcar** las siguientes casillas:
 - Hyper-V
 - Plataforma de hipervisor de Windows
 - Plataforma de máquina virtual
 4. Tras reiniciar el equipo, el comando minikube start funcionó correctamente.

Problema 3: imagepullbackoff en la parte II

- **Comando ejecutado:** kubectl apply -f deploy-mi-app-php.yaml
- **Error:** Al verificar los pods con kubectl get pods, el nuevo pod mi-app-php-deployment... Mostraba el estado imagepullbackoff.
- **Diagnóstico:** Kubernetes intentaba descargar (pull) la imagen mi-app-php:latest desde un registro público (como Docker Hub), fallando porque la imagen solo existía localmente en Minikube.
- **Solución:** Se editó el archivo deploy-mi-app-php.yaml y se añadió la política imagepullpolicy: Never en la especificación del contenedor. Esto le indicó a Kubernetes que no intentara descargar la imagen y usara la versión local disponible. Al re-aplicar el archivo YAML, el pod se inició correctamente.

