# Desplegar una APP sobre Kubernetes

**Definicion:**
Es un sistema de código libre para la automatización del despliegue, ajuste de escala y manejo de aplicaciones en contenedores.
**Objetivos:**
Implementar y desplegar una aplicación sobre Kubernetes.
**Guías:**
(*) Emplea el proyecto Laravel que se plantea en la guía:
https://github.com/learnk8s/laravel-kubernetes-demo
Profesor clone un proyecto laravel mío porque ese repositorio me daba muchos problemas. (Capturas de los problemas al final del documento).
**Trabajo por hacer y entregas:**

Hay que hacer y entregar:

1) Documentar:

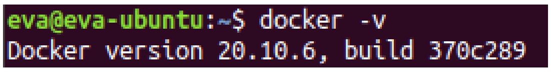
1.1) Documente paso a paso el procedimiento que ha seguido

## ❖ Requisitos para desplegar una APP sobre Kubernetes en Ubuntu:

Necesitará tener las siguientes herramientas instaladas en tu computadora:

### Docker

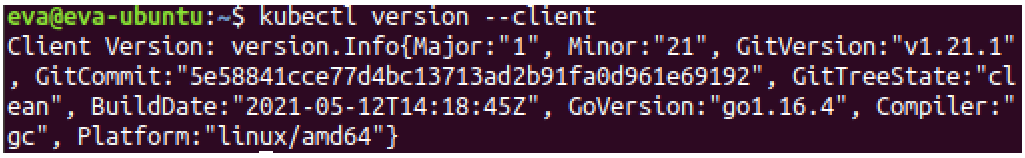
- Guia de instalacion https://docs.docker.com/compose/install/



```
eva@eva-ubuntu:~$ docker -v
Docker version 20.10.6, build 370c289
```

### kubectl

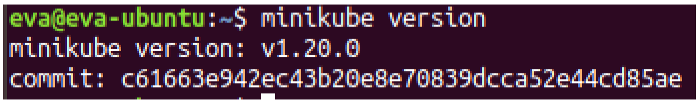
- Guia de instalacion del cliente https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/#install-kubectl-binary-with-curl-on-linux



```
eva@eva-ubuntu:~$ kubectl version --client
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.1"
, GitCommit:"5e58841cce77d4bc13713ad2b91fa0d961e69192", GitTreeState:"cl
ean", BuildDate:"2021-05-12T14:18:45Z", GoVersion:"go1.16.4", Compiler:"
gc", Platform:"linux/amd64"}
```

## Minikube

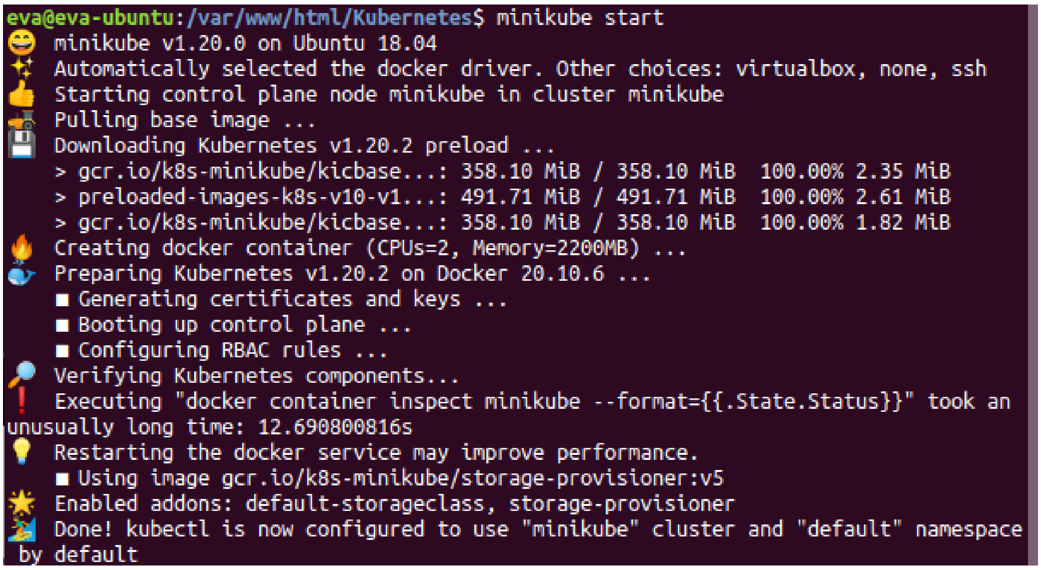
- Guia de instalacion del cluster kubernet https://cursosdedesarrollo.com/2020/07/instalacion-de-kubernetes-minikube-en-ubuntu-20-04/



```
eva@eva-ubuntu:~$ minikube version
minikube version: v1.20.0
commit: c61663e942ec43b20e8e70839dcca52e44cd85ae
```

## ❖ Arrancar minikube

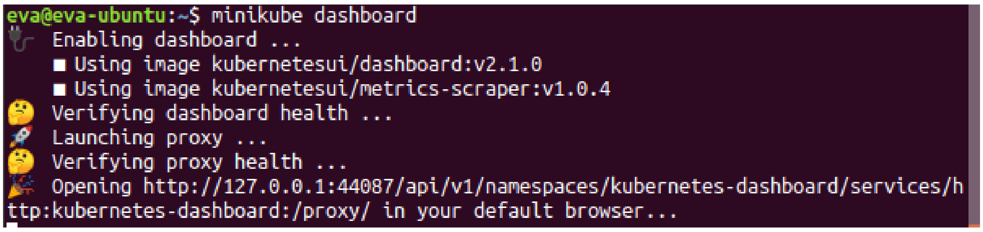
Este comando crea y configura una máquina virtual que ejecuta un clúster de Kubernetes de un solo nodo. Este comando también configura su instalación de kubectl para comunicarse con este clúster. $minikube start



```
eva@eva-ubuntu:/var/www/html/Kubernetes$ minikube start
😄  minikube v1.20.0 on Ubuntu 18.04
✨  Automatically selected the docker driver. Other choices: virtualbox, none, ssh
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
💾  Downloading Kubernetes v1.20.2 preload ...
    > gcr.io/k8s-minikube/kicbase...: 358.10 MiB / 358.10 MiB  100.00% 2.35 MiB
    > preloaded-images-k8s-v10-v1...: 491.71 MiB / 491.71 MiB  100.00% 2.61 MiB
    > gcr.io/k8s-minikube/kicbase...: 358.10 MiB / 358.10 MiB  100.00% 1.82 MiB
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.20.2 on Docker 20.10.6 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔎  Verifying Kubernetes components...
❗  Executing "docker container inspect minikube --format={{.State.Status}}" took an
unusually long time: 12.690800816s
💡  Restarting the docker service may improve performance.
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace
 by default
```

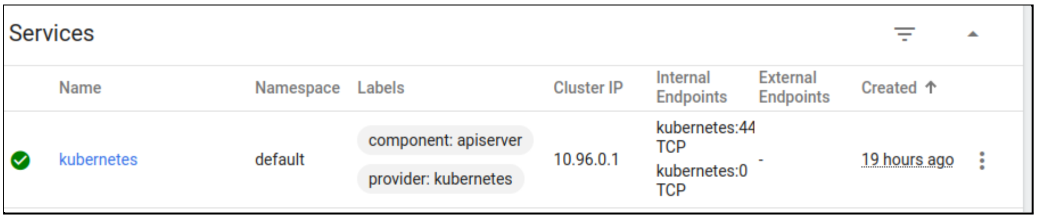
## ❖ Funcion Dashboard de minikube

Visión global del estado del cluster incluso permite desplegar aplicaciones desde una interfaz web.Para activarlo hazlo de la siguiente manera: $ minikube dashboard



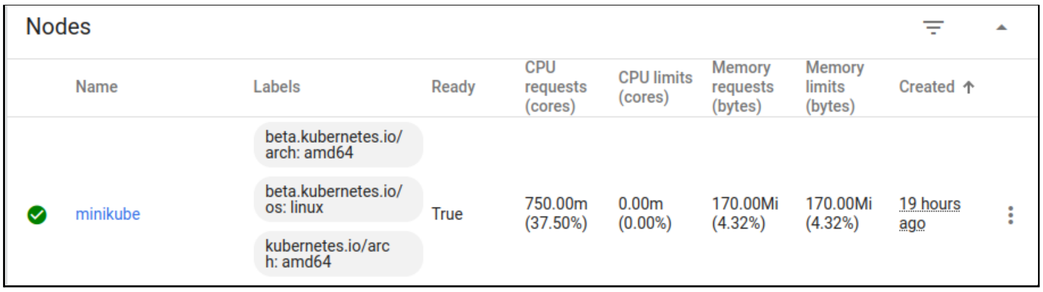
```
eva@eva-ubuntu:~$ minikube dashboard
🔌  Enabling dashboard ...
    ▪ Using image kubernetesui/dashboard:v2.1.0
    ▪ Using image kubernetesui/metrics-scraper:v1.0.4
🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:44087/api/v1/namespaces/kubernetes-dashboard/services/h
ttp:kubernetes-dashboard:/proxy/ in your default browser...
```

Nos dirige a la web donde nos muestra los servicios en funcionamiento, los nodes, pods(aplicaciones o deployments) que pueden ser un solo contenedor o un conjunto de contenedores relacionados entre sí.



Services

| | Name | Namespace | Labels | Cluster IP | Internal Endpoints | External Endpoints | Created ↑ |
|---|---|---|---|---|---|---|---|
| ✓ | kubernetes | default | component: apiserver<br>provider: kubernetes | 10.96.0.1 | kubernetes:44 TCP<br>kubernetes:0 TCP | - | 19 hours ago |



Nodes

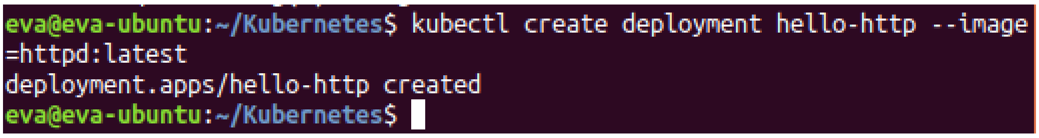
| | Name | Labels | Ready | CPU requests (cores) | CPU limits (cores) | Memory requests (bytes) | Memory limits (bytes) | Created ↑ |
|---|---|---|---|---|---|---|---|---|
| ✓ | minikube | beta.kubernetes.io/arch: amd64<br>beta.kubernetes.io/os: linux<br>kubernetes.io/arch: amd64 | True | 750.00m (37.50%) | 0.00m (0.00%) | 170.00Mi (4.32%) | 170.00Mi (4.32%) | 19 hours ago |

## ❖ Despliegue de aplicaciones

### 1. Imagen httpd:latest

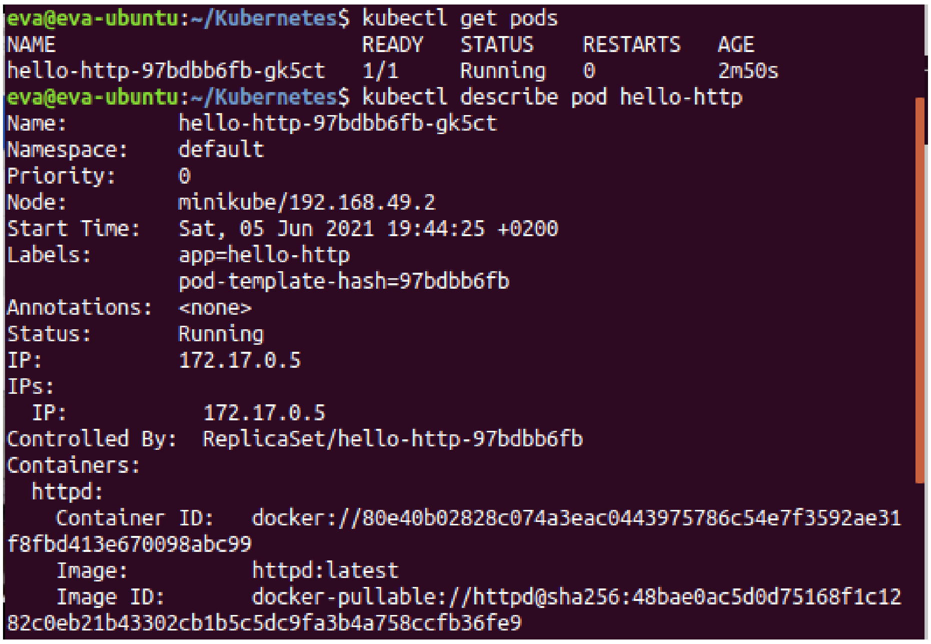
Vamos a desplegar un contenedor usando la imagen de apache (httpd:lastest), que nos descargara del hub de docker:

$ kubectl create deployment hello-http —image=httpd:latest



```
eva@eva-ubuntu:~/Kubernetes$ kubectl create deployment hello-http --image
=httpd:latest
deployment.apps/hello-http created
eva@eva-ubuntu:~/Kubernetes$ 
```

Mostrar el despliegue realizado con $kubectl get pods y para describir el pod $kubectl describe pod hello-http donde encontramos el nombre, el node , ip, los contenedores que tienen y los evento que se realizaron para hacer el despliegue



```
eva@eva-ubuntu:~/Kubernetes$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
hello-http-97bdbb6fb-gk5ct    1/1     Running   0          2m50s
eva@eva-ubuntu:~/Kubernetes$ kubectl describe pod hello-http
Name:         hello-http-97bdbb6fb-gk5ct
Namespace:    default
Priority:     0
Node:         minikube/192.168.49.2
Start Time:   Sat, 05 Jun 2021 19:44:25 +0200
Labels:       app=hello-http
              pod-template-hash=97bdbb6fb
Annotations:  <none>
Status:       Running
IP:           172.17.0.5
IPs:
  IP:           172.17.0.5
Controlled By:  ReplicaSet/hello-http-97bdbb6fb
Containers:
  httpd:
    Container ID:   docker://80e40b02828c074a3eac0443975786c54e7f3592ae31
f8fbd413e670098abc99
    Image:          httpd:latest
    Image ID:       docker-pullable://httpd@sha256:48bae0ac5d0d75168f1c12
82c0eb21b43302cb1b5c5dc9fa3b4a758ccfb36fe9
```

## Exponer servicios

Nuestros despliegues serán solo visibles desde dentro del cluster, si queremos dar visibilidad a nuestros despliegues deberemos hacerlo así:

$kubectl expose deployment hello-http --type=LoadBalancer --port=80

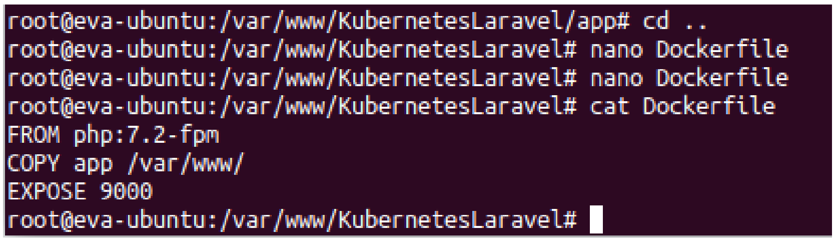
Donde le decimos a Kubernetes que exponga el puerto 80 del servicio 'hello-http' usando balanceo de carga. Podemos ver los servicios con $kubectl get services

## 2. Despliegue de la aplicación laravel

Preparar nuestra aplicación y probar su funcionamiento:

--git clone https://github.com/evaandrea22/M07-UF4-Oauth2.git app
--Ejecutar $composer install
--Añadir el archivo .env y configurarlo
--Ejecutar el comando $php artisan serve
--Generar la key $php artisan key:generate o hacerlo desde la web
--Colocar en el navegador http://127.0.0.1:8000/

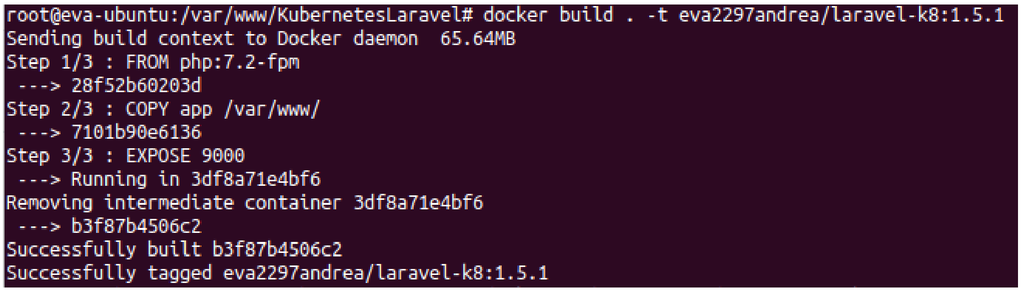
Luego ejecutar $minikube start .
Ahora para crear una imagen de Docker de su aplicación, crea el archivo Dockerfile fuera de la carpeta app en la que se encuentra el proyecto y ejecuta el comando $sudo docker build . -t eva2297andrea/laravel-k8:1.5.1



```
root@eva-ubuntu:/var/www/KubernetesLaravel/app# cd ..
root@eva-ubuntu:/var/www/KubernetesLaravel# nano Dockerfile
root@eva-ubuntu:/var/www/KubernetesLaravel# nano Dockerfile
root@eva-ubuntu:/var/www/KubernetesLaravel# cat Dockerfile
FROM php:7.2-fpm
COPY app /var/www/
EXPOSE 9000
root@eva-ubuntu:/var/www/KubernetesLaravel# 
```

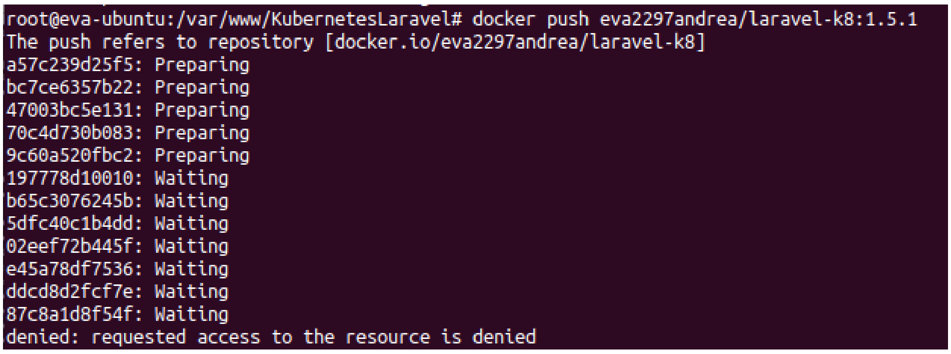
**laravel-k8:1.5.1** es el nombre ("etiqueta") de su contenedor y **"."** es la ubicación del Dockerfile y el código de la aplicación; en este caso, es el directorio actual
Luego ejecuta:

$docker build . -t eva2297andrea/laravel-k8:1.5.1 y $docker push eva2297andrea/laravel-k8:1.5.1



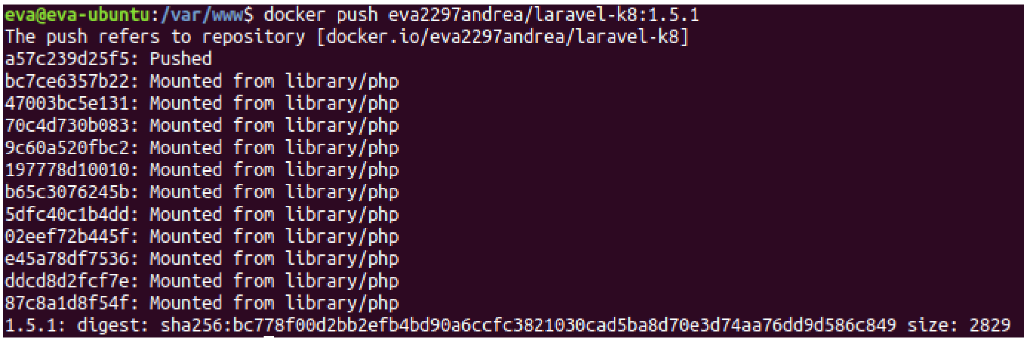
```
root@eva-ubuntu:/var/www/KubernetesLaravel# docker build . -t eva2297andrea/laravel-k8:1.5.1
Sending build context to Docker daemon  65.64MB
Step 1/3 : FROM php:7.2-fpm
 ---> 28f52b60203d
Step 2/3 : COPY app /var/www/
 ---> 7101b90e6136
Step 3/3 : EXPOSE 9000
 ---> Running in 3df8a71e4bf6
Removing intermediate container 3df8a71e4bf6
 ---> b3f87b4506c2
Successfully built b3f87b4506c2
Successfully tagged eva2297andrea/laravel-k8:1.5.1
```

Tuve un problema con docker push y la solución era salir de sudo, loguearse en docker nuevamente y ejecutar el comando nuevamente



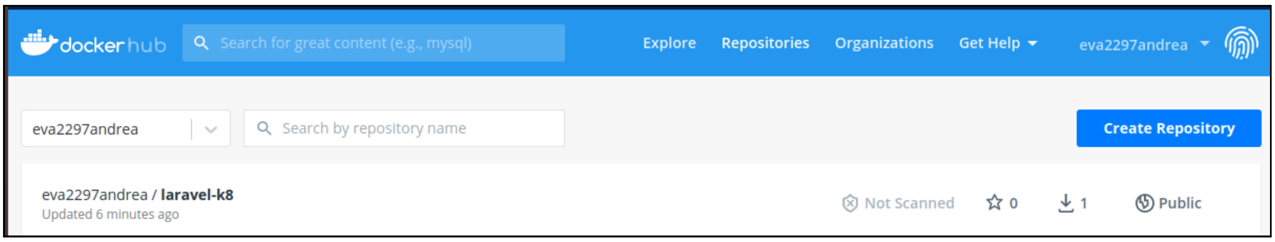
```
root@eva-ubuntu:/var/www/KubernetesLaravel# docker push eva2297andrea/laravel-k8:1.5.1
The push refers to repository [docker.io/eva2297andrea/laravel-k8]
a57c239d25f5: Preparing
bc7ce6357b22: Preparing
47003bc5e131: Preparing
70c4d730b083: Preparing
9c60a520fbc2: Preparing
197778d10010: Waiting
b65c3076245b: Waiting
5dfc40c1b4dd: Waiting
02eef72b445f: Waiting
e45a78df7536: Waiting
ddcd8d2fcf7e: Waiting
87c8a1d8f54f: Waiting
denied: requested access to the resource is denied
```

solucion:[https://forums.docker.com/t/docker-push-error-requested-access-to-the-resource-is-denied/64468](https://forums.docker.com/t/docker-push-error-requested-access-to-the-resource-is-denied/64468)



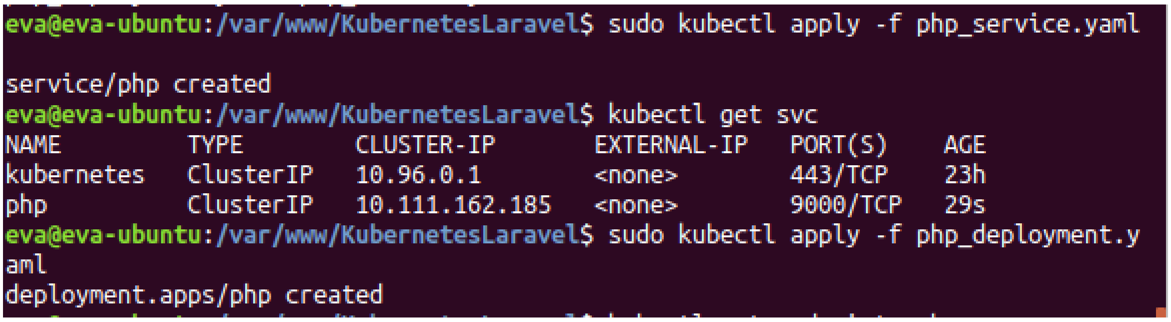
```
eva@eva-ubuntu:/var/www$ docker push eva2297andrea/laravel-k8:1.5.1
The push refers to repository [docker.io/eva2297andrea/laravel-k8]
a57c239d25f5: Pushed
bc7ce6357b22: Mounted from library/php
47003bc5e131: Mounted from library/php
70c4d730b083: Mounted from library/php
9c60a520fbc2: Mounted from library/php
197778d10010: Mounted from library/php
b65c3076245b: Mounted from library/php
5dfc40c1b4dd: Mounted from library/php
02eef72b445f: Mounted from library/php
e45a78df7536: Mounted from library/php
ddcd8d2fcf7e: Mounted from library/php
87c8a1d8f54f: Mounted from library/php
1.5.1: digest: sha256:bc778f00d2bb2efb4bd90a6ccfc3821030cad5ba8d70e3d74aa76dd9d586c849 size: 2829
```

En nuestro docker hub podemos ver el resultado:

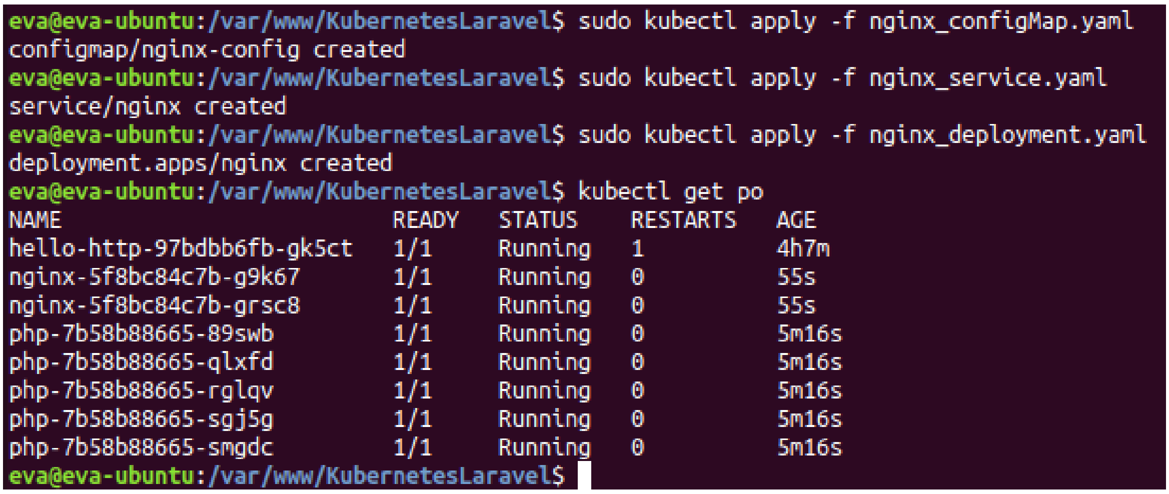


eva2297andrea / **laravel-k8**
Updated 6 minutes ago

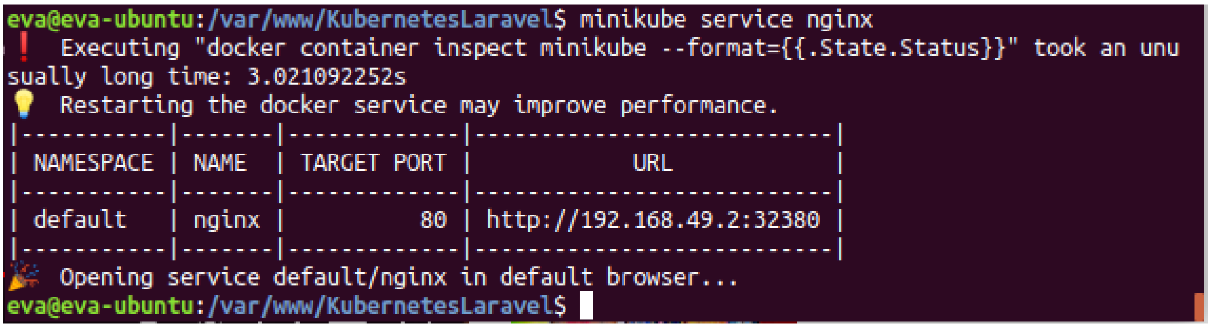
ahora implemente esta aplicación en el clúster de kubernetes:



```
eva@eva-ubuntu:/var/www/KubernetesLaravel$ sudo kubectl apply -f php_service.yaml

service/php created
eva@eva-ubuntu:/var/www/KubernetesLaravel$ kubectl get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP    23h
php          ClusterIP   10.111.162.185   <none>        9000/TCP   29s
eva@eva-ubuntu:/var/www/KubernetesLaravel$ sudo kubectl apply -f php_deployment.y
aml
deployment.apps/php created
```

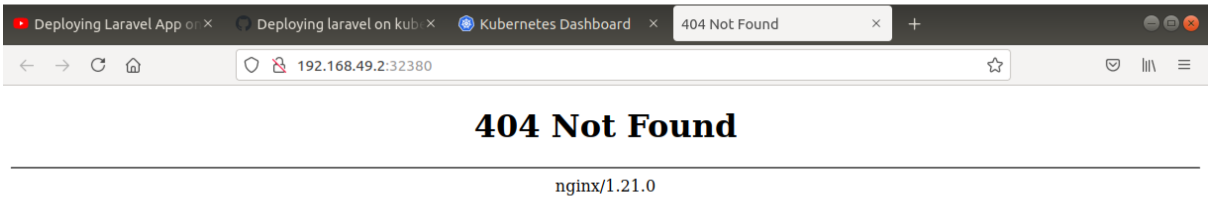


```
eva@eva-ubuntu:/var/www/KubernetesLaravel$ sudo kubectl apply -f nginx_configMap.yaml
configmap/nginx-config created
eva@eva-ubuntu:/var/www/KubernetesLaravel$ sudo kubectl apply -f nginx_service.yaml
service/nginx created
eva@eva-ubuntu:/var/www/KubernetesLaravel$ sudo kubectl apply -f nginx_deployment.yaml
deployment.apps/nginx created
eva@eva-ubuntu:/var/www/KubernetesLaravel$ kubectl get po
NAME                          READY   STATUS    RESTARTS   AGE
hello-http-97bdbb6fb-gk5ct    1/1     Running   1          4h7m
nginx-5f8bc84c7b-g9k67        1/1     Running   0          55s
nginx-5f8bc84c7b-grsc8        1/1     Running   0          55s
php-7b58b88665-89swb          1/1     Running   0          5m16s
php-7b58b88665-qlxfd          1/1     Running   0          5m16s
php-7b58b88665-rglqv          1/1     Running   0          5m16s
php-7b58b88665-sgj5g          1/1     Running   0          5m16s
php-7b58b88665-smgdc          1/1     Running   0          5m16s
eva@eva-ubuntu:/var/www/KubernetesLaravel$
```



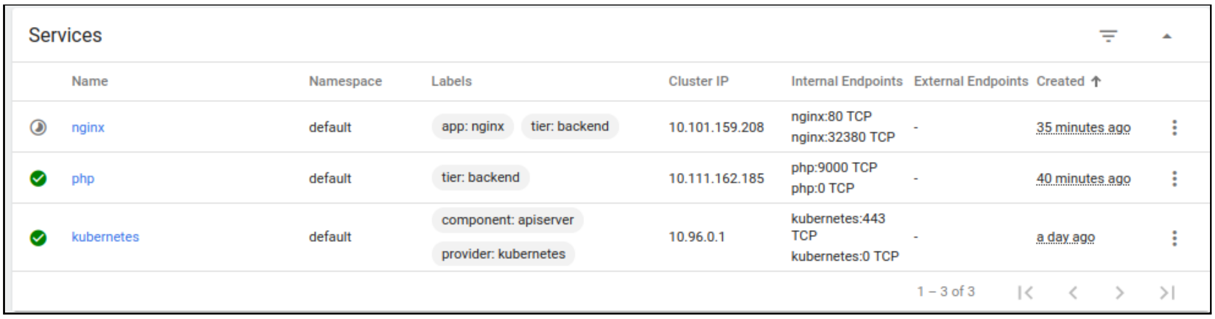
```
eva@eva-ubuntu:/var/www/KubernetesLaravel$ minikube service nginx
❗  Executing "docker container inspect minikube --format={{.State.Status}}" took an unu
sually long time: 3.021092252s
💡  Restarting the docker service may improve performance.
|-----------|-------|-------------|---------------------------|
| NAMESPACE | NAME  | TARGET PORT |            URL            |
|-----------|-------|-------------|---------------------------|
| default   | nginx |          80 | http://192.168.49.2:32380 |
|-----------|-------|-------------|---------------------------|
🎉  Opening service default/nginx in default browser...
eva@eva-ubuntu:/var/www/KubernetesLaravel$
```

Al ejecutar el comando se ha de abrir la web laravel creada pero me da el error que no es encontrado



404 Not Found

nginx/1.21.0



Services

| | Name | Namespace | Labels | Cluster IP | Internal Endpoints | External Endpoints | Created ↑ |
|---|---|---|---|---|---|---|---|
| ◔ | nginx | default | app: nginx<br>tier: backend | 10.101.159.208 | nginx:80 TCP<br>nginx:32380 TCP | - | 35 minutes ago |
| ✓ | php | default | tier: backend | 10.111.162.185 | php:9000 TCP<br>php:0 TCP | - | 40 minutes ago |
| ✓ | kubernetes | default | component: apiserver<br>provider: kubernetes | 10.96.0.1 | kubernetes:443 TCP<br>kubernetes:0 TCP | - | a day ago |

1 – 3 of 3

1.2) Pruebe la aplicación

1.4) Grabar un vídeo con evidencias del funcionamiento

1.5) Bibliografía

Otra guia para desplegar la app en kubernetes

https://www.youtube.com/watch?v=8e0vtNO-T_I&t=560s

https://www.digitalocean.com/community/tutorials/how-to-deploy-laravel-7-and-mysql-on-kubernetes-using-helm-es

Segui la guia https://learnk8s.io/blog/kubernetes-deploy-laravel-the-easy-way para descargar e instalar la aplicación que desplegaremos sobre Kubernetes

Problemas al ejecutar composer install: https://stackoverflow.com/questions/65454412/root-composer-json-requires-php-7-3-but-your-php-version-8-0-0-does-not-satis

2) URLs:

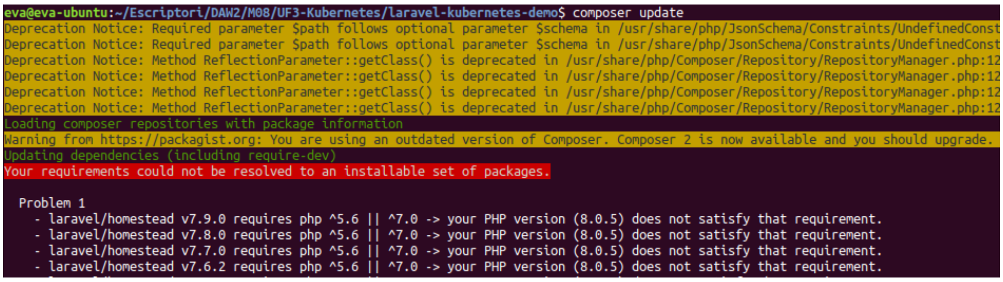
2.1) URL en repo de GitHub que contiene su código

https://github.com/evaandrea22/M08-UF3-Desplegar_AppLaravel_onKubernetes

2.2) URL de la web de su proyecto

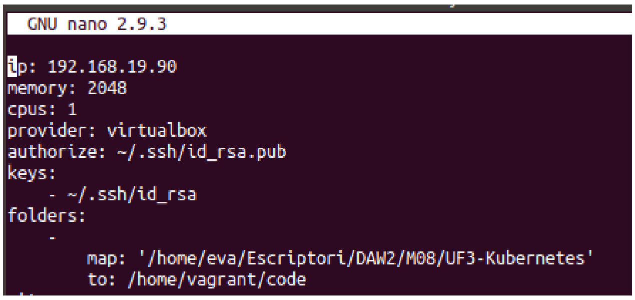
Problemas en el desarrollo de la práctica:

Después de clonar el repositorio y al momento de ejecutar composer install me daba muchos problemas.

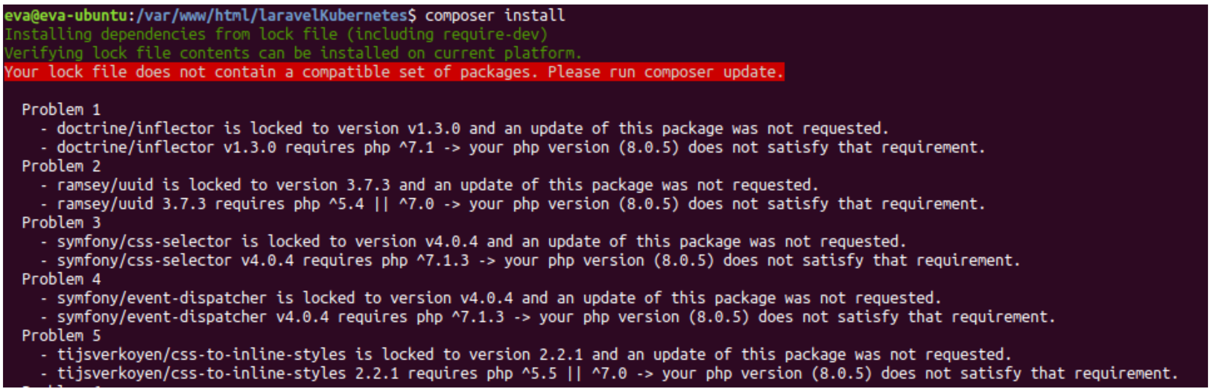


```
eva@eva-ubuntu:~/Escriptori/DAW2/M08/UF3-Kubernetes/laravel-kubernetes-demo$ composer update
Deprecation Notice: Required parameter $path follows optional parameter $schema in /usr/share/php/JsonSchema/Constraints/UndefinedConst
Deprecation Notice: Required parameter $path follows optional parameter $schema in /usr/share/php/JsonSchema/Constraints/UndefinedConst
Deprecation Notice: Method ReflectionParameter::getClass() is deprecated in /usr/share/php/Composer/Repository/RepositoryManager.php:12
Deprecation Notice: Method ReflectionParameter::getClass() is deprecated in /usr/share/php/Composer/Repository/RepositoryManager.php:12
Deprecation Notice: Method ReflectionParameter::getClass() is deprecated in /usr/share/php/Composer/Repository/RepositoryManager.php:12
Deprecation Notice: Method ReflectionParameter::getClass() is deprecated in /usr/share/php/Composer/Repository/RepositoryManager.php:12
Loading composer repositories with package information
Warning from https://packagist.org: You are using an outdated version of Composer. Composer 2 is now available and you should upgrade.
Updating dependencies (including require-dev)
Your requirements could not be resolved to an installable set of packages.

  Problem 1
    - laravel/homestead v7.9.0 requires php ^5.6 || ^7.0 -> your PHP version (8.0.5) does not satisfy that requirement.
    - laravel/homestead v7.8.0 requires php ^5.6 || ^7.0 -> your PHP version (8.0.5) does not satisfy that requirement.
    - laravel/homestead v7.7.0 requires php ^5.6 || ^7.0 -> your PHP version (8.0.5) does not satisfy that requirement.
    - laravel/homestead v7.6.2 requires php ^5.6 || ^7.0 -> your PHP version (8.0.5) does not satisfy that requirement.
```

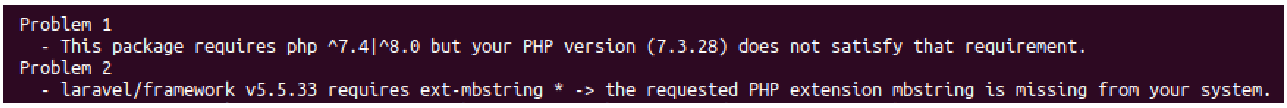
Solucion sudo nano Homestead.yaml



```
  GNU nano 2.9.3

ip: 192.168.19.90
memory: 2048
cpus: 1
provider: virtualbox
authorize: ~/.ssh/id_rsa.pub
keys:
    - ~/.ssh/id_rsa
folders:
    -
        map: '/home/eva/Escriptori/DAW2/M08/UF3-Kubernetes'
        to: /home/vagrant/code
```



```
eva@eva-ubuntu:/var/www/html/laravelKubernetes$ composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Your lock file does not contain a compatible set of packages. Please run composer update.

  Problem 1
    - doctrine/inflector is locked to version v1.3.0 and an update of this package was not requested.
    - doctrine/inflector v1.3.0 requires php ^7.1 -> your php version (8.0.5) does not satisfy that requirement.
  Problem 2
    - ramsey/uuid is locked to version 3.7.3 and an update of this package was not requested.
    - ramsey/uuid 3.7.3 requires php ^5.4 || ^7.0 -> your php version (8.0.5) does not satisfy that requirement.
  Problem 3
    - symfony/css-selector is locked to version v4.0.4 and an update of this package was not requested.
    - symfony/css-selector v4.0.4 requires php ^7.1.3 -> your php version (8.0.5) does not satisfy that requirement.
  Problem 4
    - symfony/event-dispatcher is locked to version v4.0.4 and an update of this package was not requested.
    - symfony/event-dispatcher v4.0.4 requires php ^7.1.3 -> your php version (8.0.5) does not satisfy that requirement.
  Problem 5
    - tijsverkoyen/css-to-inline-styles is locked to version 2.2.1 and an update of this package was not requested.
    - tijsverkoyen/css-to-inline-styles 2.2.1 requires php ^5.5 || ^7.0 -> your php version (8.0.5) does not satisfy that requirement.
```



```
  Problem 1
    - This package requires php ^7.4|^8.0 but your PHP version (7.3.28) does not satisfy that requirement.
  Problem 2
    - laravel/framework v5.5.33 requires ext-mbstring * -> the requested PHP extension mbstring is missing from your system.
```

Soluciones que vi por internet pero no me funcionaron:

https://askubuntu.com/questions/858722/problem-on-laravel-composer-installation-in-12-04

https://stackoverflow.com/questions/65454412/root-composer-json-requires-php-7-3-but-your-php-version-8-0-0-does-not-satis