# Curso k8s Taller semana 1

## Laboratorio 1

## Prerrequisitos

Tener instalado Docker

https://docs.docker.com/docker-for-windows/install/

## Primer punto (refuerzo)

Ejecute el siguiente comando “docker container run --publish 8080:80 -d --name webserver nginx” y verifique:

* Que la versión de Docker esté instalada y sea igual o superior a 20.10.5 usando el comando “docker versión”
* Que liste las imágenes y visualice la imagen que acaba de crear con el comando “docker image ls”
* Que liste los contenedores que se están ejecutando, usando el comando “docker container ls”
* Que liste todos los contenedores que están en su máquina usando “docker container ls -a”
* Que pueda acceder al contenedor usando el comando “docker container exec -it webserver bash”
  1. Verifique la distribución de Linux usando el comando “cat /etc/\*-release”
* Que pueda ver su número de proceso (Los contenedores son procesos de la máquina)
  1. docker run -it --rm --privileged --pid=host justincormack/nsenter1
  2. ps aux
* Que pueda acceder desde su navegador usando el puerto especificado
* Que pueda acceder a sus logs usando el comando “docker container logs webserver”
* Verifique los recursos físicos usados por el contenedor usando el comando “docker container stats”

## Segundo punto

Usted, como Cloud Engineer, ha sido asignado para migrar una aplicación realizada en Django a contenedores. Esta aplicación deberá cumplir los siguientes requisitos:

* Versión de Python 3.9 o superior.
* Se deberán entregar las evidencias de que la aplicación se ejecute satisfactoriamente.

## Solución

* Descargue el repositorio [git@github.com:felipe-devops/django-template.git](mailto:git@github.com:felipe-devops/django-template.git)
* Ingrese a la carpeta django-template/container\_webapp
* Visualice el archivo Dockerfile: este indica cuál será la imagen base a usar, tendrá el directorio raíz de la aplicación, instalará las dependencias y moverá la aplicación a la carpeta seleccionada.
* Ejecute el comando docker network create django\_app
* Ejecute el comando “docker build --tag django-app .” que construirá la imagen
* Ejecute el comando “docker container run --publish 8084:8080 -d --name django\_app --network django\_app django-app” que construirá un contenedor con la aplicación y la expondrá externamente hacia el puerto 8484.
* Verifique en su navegador que aparezca la página de inicio de un proyecto de Django.
* Revise los logs de inicio “docker logs django\_webapp”
* Ingrese al contenedor “docker container exec -it django\_webapp bash” y verifique qué se encuentra en la carpeta “/code”
* Salga del contenedor “exit”
* Verifique el proceso que se está ejecutando:
  + docker run -it --rm --privileged --pid=host justincormack/nsenter1
* Detenga el contenedor “docker container stop django\_webapp”
* Borre el contenedor “docker container rm django\_webapp”
* Borre la imagen “docker image rm django-app”

Revisar un poco sobre las redes

Crear una red

docker network create django\_app

docker container run --name some-postgres --network django\_app -e POSTGRES\_PASSWORD=mysecretpassword -d postgres

docker network inspect django\_app

Usar un contenedor en una red particular

docker container run -it --network django\_app ubuntu

## Taller

El objetivo es hacer algo muy similar al taller expuesto en la sesión pero con una aplicación diferente. Para esto, encontramos en la página oficial de Docker cómo crear imágenes y ejecutar contenedores usando Node.js. el objetivo es seguir las guías de los siguientes enlaces:

1. <https://docs.docker.com/language/nodejs/build-images/>
2. <https://docs.docker.com/language/nodejs/run-containers/>