


DESAFIO N° 16-17
BOTCAMP DE DEVOPS ENGINEER

ALEXIS ROLDAN – FLORENCIA
PEZZELLA – MAURO USAY

educación IT


Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

INDICE

Objetivo	2
Desafío:	3
Parte 1: Vagrant	3
Parte 2: Docker	3
Parte 3: Automatización de Dockerfile	3
Parte 4: Kubernetes	4
Parte 5: Helm y ArgoCD	4
Entregable	5
Respuestas	6
Vagrant	6
Docker	9
Automatizacion de Dockerfile	12
Kubernetes	15
Helm y ArgoCD	20
Entregable	31


Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

Objetivo

El objetivo de este desafío es realizar un trabajo que integre lo visto en los distintos módulos de esta fase. El desafío será grupal (grupos de 2 a 5 personas) y tendrán que dividirse las tareas para lograr hacer todo.

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1

Desafío:

Prerrequisitos:

- Repositorio de código (GitHub, gitlab, bitbucket, etc) a elección del grupo
- Grupo 2 a 5 personas
- Decidir qué aplicación o conjunto de aplicaciones utilizaran para el desafío

Parte 1: Vagrant

1) Durante esta parte del desafío, tendrán que crear una máquina en Vagrant donde instalarán todas las herramientas utilizadas en la práctica, lo recomendable es que trabajen de forma tal que puedan ir agregando herramientas a medida que las van necesitando (docker, kubectl, minikube, microk8s, kind o cualquier otra herramienta utilizada para crear un clúster de kubernetes).

2) Para configurar la máquina de vagrant podrán usar el provider que más les guste, incluso el de Ansible.

3) Tengan en cuenta que el objetivo de esta máquina virtual es el de poder replicar absolutamente todo el desafío de una forma sencilla a cualquier persona que tenga acceso al repositorio, recomendamos seguir la modalidad utilizada en el módulo 16 para la parte de kubernetes.

Parte 2: Docker

1) Durante esta parte del desafío tendrán que instalar las herramientas necesarias en la máquina virtual que les permita Dockerizar aplicaciones


2) Habiendo elegido la aplicación o conjunto de aplicaciones que van a dockerizar, procederán a crear los Dockerfile y Docker compose necesarios (de forma manual)

Parte 3: Automatización de Dockerfile

1) Deberán automatizar la creación del Dockerfile/Docker compose utilizando alguna herramienta de CI/CD a elección, el objetivo es que cualquier cambio realizado en el código (ya sea en el Dockerfile/Dockercompose o cambios en el código de la aplicación) active un pipeline que les ejecute un docker build y suba esa imagen a DockerHub, para esto pueden utilizar Jenkins, Gitlab CI/CD, GitHub Actions, etc. La recomendación es utilizar la herramienta que venga con la herramienta de repositorios que hayan decidido utilizar (por ejemplo, GitHub Actions).

2) Como parte de esta automatización, tendrán que si o si incluir algún tipo de linter, revisión de vulnerabilidades y algún tipo de testeo básico de la aplicación de algún tipo.

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1

Parte 4: Kubernetes

1) Ya habiendo containerizado las distintas partes de la aplicación y automatizado la construcción de la imagen, toca el turno de convertir estos contenedores en recursos de Kubernetes. Los mismos deberán ser deployments + otros recursos adicionales (secretos, volúmenes, configmaps, etc), la configuración del deployment debe permitir que siempre que haya una nueva versión, se cree 1 pod del nuevo replicaset antes de eliminar un pod del replicaset anterior, esto garantizara que solamente una vez que el pod de la nueva versión esté funcionando, se remueva un pod de la versión anterior.

2) Para exponer la aplicación y poder acceder a ella podremos usar el port-forward de kubectl junto al puente de vagrant ssh, aunque pueden buscar otras maneras de exponer la aplicación en caso de requerirlo.

Parte 5: Helm y ArgoCD


1) La primera etapa consiste en convertir los distintos recursos que creamos en Kubernetes para nuestra aplicación en un paquete de Helm, no solo tendremos que convertirlo sino que también tendremos que convertir estos recursos en templates junto a sus variables para poder realizar modificaciones y reutilizar los recursos definidos en el template.

2) Una vez ya convertida la aplicación en un template de Kubernetes, tendrán que levantar una instancia de ArgoCD, conectarla a su clúster de Kubernetes y:

a) Levantar una aplicación que apunte a su aplicación convertida en un chart de Helm, donde le configurarán los valores y tendrán que configurar la aplicación de Argo para que tenga el Self healing habilitado como así también el autosync

b) Levantar una aplicación que apunte a su aplicación en recursos de Kubernetes normales (sin Helm), con sync manual y sin self healing, tengan en cuenta que para que ambas aplicaciones puedan convivir tendrán que ser desplegadas en distintos namespaces para evitar repetición de nombres.

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

Entregable

Los entregables serán almacenados en la carpeta compartida que tienen en drive con el formato (<carpeta con su nombre>/<Fase>/<módulo>/archivo). Por poner un ejemplo, el instructivo se almacenaría en la carpeta compartida con el nombre Zdenko Hraste, en una carpeta llamada Fase 3 que dentro tendrá otra carpeta llamada Módulo 16-17 y que dentro almacenare todos los archivos relevantes a este desafío.

De la misma forma que en el desafío anterior, tendrán que agregar los archivos a su carpeta en drive dentro de la carpeta Módulo 16-17 dentro de la Fase 3.

Se esperan los archivos:

- Instructivo
- Link al repositorio de código (Documentado en el instructivo)
- Link al repositorio de la imagen (Documentado en el instructivo)
- Una carpeta con una copia del repositorio de código (código-M16-17)


El repositorio deberá seguir una estructura similar a las partes del desafío, pueden utilizar como referencia el siguiente repositorio:

<https://github.com/zdenkotraste/workshop-nerdearla/tree/main>

Si bien en el instructivo documentaran todo, es importante que trabajen el Readme del repositorio (incluso pueden agregar un Readme por carpeta con más información para probar cada parte del mismo). El objetivo es que una persona X pueda agarrar su repositorio y probarlo de punta a punta (es decir, probar todo en conjunto y también poder probar cada una de las partes por separado).

Recuerden seguir las instrucciones al pie de la letra para los entregables.

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1

Respuestas

Vagrant

- 1- Para la creación del entorno de Vagrant, debemos crear una carpeta y ejecutar la inicialización de vagrant con un "init"

\$ vagrant init ubuntu/focal64


```
Vagrantfile
1  : -*- mode: ruby -*-
2  : vi: set ft=ruby :
3
4  : All Vagrant configuration is done below. The "2" in Vagrant.configure
5  : configures the configuration version (we support older styles for
6  : backwards compatibility). Please don't change it unless you know what
7  : you're doing.
8  'agrant.configure("2") do |config|
9    # The most common configuration options are documented and commented below.
10   # For a complete reference, please see the online documentation at
11   # https://docs.vagrantup.com.
12
13   # Every Vagrant development environment requires a box. You can search for
14   # boxes at https://vagrantcloud.com/search.
15   config.vm.box = "ubuntu/focal64"
16
17   # Disable automatic box update checking. If you disable this, then
18   # boxes will only be checked for updates when the user runs
19   # `vagrant box outdated`. This is not recommended.
20   # config.vm.box_check_update = false
21
22   # Create a forwarded port mapping which allows access to a specific port
```

Una vez creado el Vagrantfile, se edita el archivo en función de los recursos que necesitemos.

```
config.vm.provider "virtualbox" do |vb|
  # # Display the VirtualBox GUI when booting the machine
  # vb.gui = true
  #
  # # Customize the amount of memory on the VM:
  vb.memory = "12288"
  vb.cpus = "8"
end
```

Una vez seteado los parámetros de los recursos de nuestra maquina virtual procederemos a crear la vm con el comando "up"

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

```

macbookpro@Flor M13_vagrant % vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'ubuntu/focal64' could not be found. Attempting to find and install...
default: Box Provider: virtualbox
default: Box Version: >= 0
==> default: Loading metadata for box 'ubuntu/focal64'
default: URL: https://vagrantcloud.com/ubuntu/focal64
==> default: Adding box 'ubuntu/focal64' (v20230922.0.0) for provider: virtualbox
default: Downloading: https://vagrantcloud.com/ubuntu/boxes/focal64/versions/20230922.0.0/providers/virtualbox/unknown/vagrant.box
Download redirected to host: cloud-images.ubuntu.com
Progress: 2% (Rate: 838k*/s, Estimated time remaining: 0:14:30)
  
```

Luego de ejecutar la creación, verificaremos el estado mediante el comando “status”

```

macbookpro@Flor M13_vagrant % vagrant status
Current machine states:

default                running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
  
```

Para acceder a nuestra máquina virtual ejecutaremos el comando “ssh”

```

macbookpro@Flor M14 % vagrant ssh
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-163-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage


System information as of Sun Nov 26 22:03:12 UTC 2023

System load:  1.11           Processes:            159
Usage of /:   6.1% of 38.7GB Users logged in:      0
Memory usage: 5%            IPv4 address for docker0: 172.17.0.1
Swap usage:   0%            IPv4 address for enp0s3: 10.0.2.15

Expanded Security Maintenance for Applications is not enabled.

42 updates can be applied immediately.
29 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
  
```


Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

Para la instalación de nuestras aplicaciones para poder realizar el trabajo practico se creo un archivo denominado Bootstrap.sh donde contiene los comandos de instalación de las distintas aplicaciones que utilizaremos


SCRIPT DE INSTALACION

```

bootstrap.sh X
Vagrant > bootstrap.sh
You, 8 hours ago | 1 author (You)
1 # Docker install
2 curl -fsSL https://get.docker.com -o get-docker.sh
3 sudo sh get-docker.sh
4
5
6 # Use docker without sudo
7 sudo groupadd docker
8 sudo usermod -aG docker vagrant
9 newgrp docker
10
11 #Install kubect1
12 curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubect1"
13 sudo install -o root -g root -m 0755 kubect1 /usr/local/bin/kubect1
14
15 #Install minikube
16 curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
17 sudo install minikube-linux-amd64 /usr/local/bin/minikube
18
19 #Install Helm
20 curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
21 chmod 700 get_helm.sh
22 ./get_helm.sh
23
24 #Install ArgoCD
25 curl -sSL -o argocd-linux-amd64 https://github.com/argoproj/argo-cd/releases/latest/download/argocd-linux-amd64
26 sudo install -m 555 argocd-linux-amd64 /usr/local/bin/argocd
27 rm argocd-linux-amd64
28

```

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1

Docker

Todo nuestro desarrollo será alojado en el repositorio de GitHub

REPOSITORIO EN GITHUB:

Se crea el repositorio en GitHub: <https://github.com/alexlpda1420/desafio16-17>

DOCKERFILE:

A continuación, se procederán a crear los Dockerfile y Docker compose necesarios (de forma manual).

Se construye un Dockerfile para ejecutar Plex Media Server. Se trata de un servicio que funciona como un servidor multimedia donde se puede reproducir todo aquello que se encuentre dentro de mi computadora, sería como mi propio Mubi/Netflix/etc.

Para eso se ejecuta el Dockerfile con las siguientes instrucciones:

```

1 # Ubuntu image
2 FROM ubuntu:latest
3
4 # Update and install package
5 RUN apt-get update && \
6     apt-get install -y curl && \
7     apt-get clean && \
8     rm -rf /var/lib/apt/lists/*
9
10 # Download and install Plex Media Server
11 RUN curl -LO "https://downloads.plex.tv/plex-media-server-new/1.32.6.7557-1cf77d501/debian/plexmediaserver_1.32.6.7557-1cf77d501_amd64.deb" && \
12     dpkg -i plexmediaserver_1.32.6.7557-1cf77d501_amd64.deb
13
14 # Add port and volumes
15 EXPOSE 32400
16 VOLUME /config
17 VOLUME /media
18
19 CMD ["/usr/lib/plexmediaserver/Plex Media Server", "-c", "/usr/sbin/start_pms", "/config", "/media"]
20

```

FROM: Se crea el docker file a partir de la imagen de Ubuntu.

RUN: Actualiza el sistema y corre los paquetes.


Descarga e instala Plex.

EXPOSE: El puerto que Plex Media Server utiliza para la comunicación es el 32400.

VOLUME: Se definen los volúmenes para guardar la configuración.

CMD: Y se inicia y ejecuta el contenedor.

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

REVISIÓN DE VULNERABILIDADES:

Antes de iniciar la dockerización se verifica el Dockerfile con la herramienta de código abierto llamada Hadolint. La misma fue creada por GitHub.



A smarter Dockerfile linter that helps you build [best practice Docker images](#). The linter is parsing the Dockerfile into an AST and performs rules on top of the AST. It additionally is using the famous [Shellcheck](#) to lint the Bash code inside `RUN` instructions. Please [help me improve the linter](#) with your suggestions.

```

8      rm -rf /var/lib/apt/lists/*
9
10 # Download and install Plex Media Server
11 RUN curl -LO "https://downloads.plex.tv/plex-media-server-new/1.32.6.7557-1cf77d501_amd64.deb"
12     dpkg -i plexmediaserver_1.32.6.7557-1cf77d501_amd64.deb
13
14 # Add port and volumes
15 EXPOSE 32400
16 VOLUME /config
17 VOLUME /media
18
19 CMD ["/usr/lib/plexmediaserver/Plex Media Server", "-c", "/usr/"]

```

Lint

Clear

Una vez que hemos verificado nuestra imagen, procederemos a crear el contenedor

EJUTANDO DOCKER PARA PLEX MEDIA SERVER:

Se ejecuta en la terminal el siguiente comando para crear la imagen llamada docker-on-plex:


`docker build . -t docker-on-plex`

```

vagrant@ubuntu-focal:~/vagrant/Dockerfiles$ docker build . -t docker-on-plex
[+] Building 1.7s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 637B
=> [internal] load .dockerignore
=> transferring context: 2B
=> [internal] load metadata for docker.io/library/ubuntu:latest
=> [1/3] FROM docker.io/library/ubuntu:latest@sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
=> CACHED [2/3] RUN apt-get update && apt-get install -y curl && apt-get clean && rm -rf /var/lib/apt/lists/*
=> CACHED [3/3] RUN curl -LO "https://downloads.plex.tv/plex-media-server-new/1.32.6.7557-1cf77d501/debian/plexmediaserver_1.32.6.7557-1cf77d501_amd64.deb"
=> exporting to image
=> exporting layers
=> writing image sha256:b1db268e25c5530e1babca76e904683cf26b91c61fb902bb7b308d8cc7f334626
=> naming to docker.io/library/docker-on-plex
vagrant@ubuntu-focal:~/vagrant/Dockerfiles$

```

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1

Para consultar si la imagen efectivamente fue creada, se indica el comando:

`docker image ls`

```
vagrant@ubuntu-focal:~/vagrant/Dockerfile$ docker image ls
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
docker-on-plex      latest      b1db268e25c5  18 hours ago  353MB
vagrant@ubuntu-focal:~/vagrant/Dockerfile$
```

Ahora se crea el contenedor mediante el siguiente comando:

`docker run -d --name plex-movies -p 32400:32400 docker-on-plex`

```
vagrant@ubuntu-focal:~/vagrant/Dockerfile$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
db0b55f2f02a  docker-on-plex "/usr/lib/plexmedias..." 15 seconds ago Up 14 seconds 0.0.0.0:32400->32400/tcp, :::32400->32400/tcp  plex-movies
vagrant@ubuntu-focal:~/vagrant/Dockerfile$
```

Y se consulta mediante el comando `docker ps` el listado de contenedores creados y ejecutando en este momento:

```
vagrant@ubuntu-focal:~/vagrant/Dockerfile$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
db0b55f2f02a  docker-on-plex "/usr/lib/plexmedias..." 2 minutes ago Up 2 minutes  0.0.0.0:32400->32400/tcp, :::32400->32400/tcp  plex-movies
vagrant@ubuntu-focal:~/vagrant/Dockerfile$
```


Para visualizar la totalidad de contenedores creados se puede consultar mediante el comando `docker ps -a`

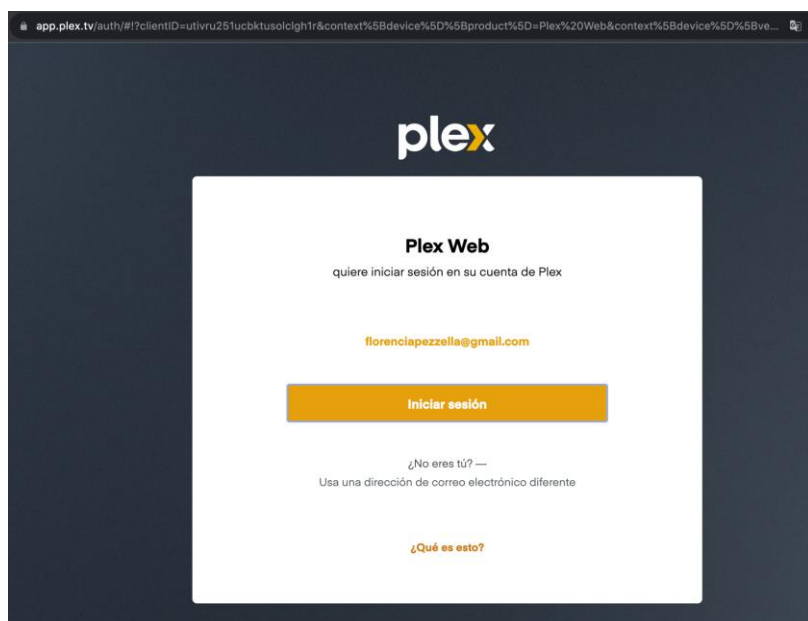
```
vagrant@ubuntu-focal:~/vagrant/Dockerfile$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
db0b55f2f02a  docker-on-plex "/usr/lib/plexmedias..." 2 minutes ago Up 2 minutes  0.0.0.0:32400->32400/tcp, :::32400->32400/tcp  plex-movies
vagrant@ubuntu-focal:~/vagrant/Dockerfile$
```

Para acceder a Plex a través del contenedor de Docker se ejecuta mediante el navegador web indicando la url: `localhost:32400/web`

```
localhost:32400/web
Plex - localhost:32400/web
localhost:32400/web - Google Search
Plex - localhost:32400/web/index.html#!/setup/391773c71fff72df75d53c30797b9325d3150eab
```

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1



URL DEL REPOSITORIO DE DOCKER:

https://hub.docker.com/repository/docker/florpezz/plex_on_docker/general

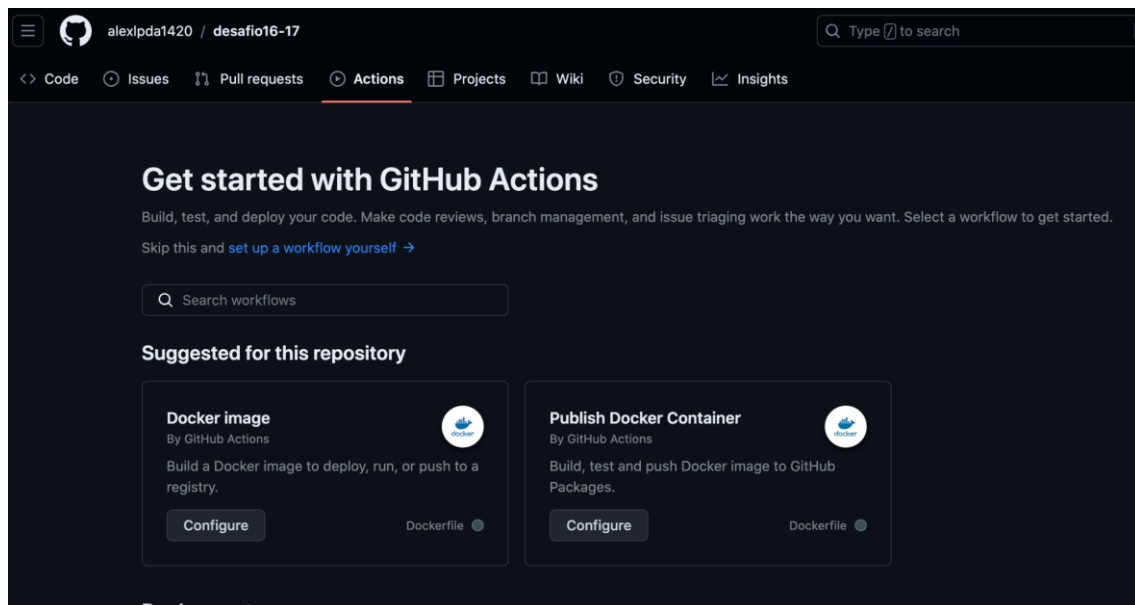
Automatización de Dockerfile

Educación IT

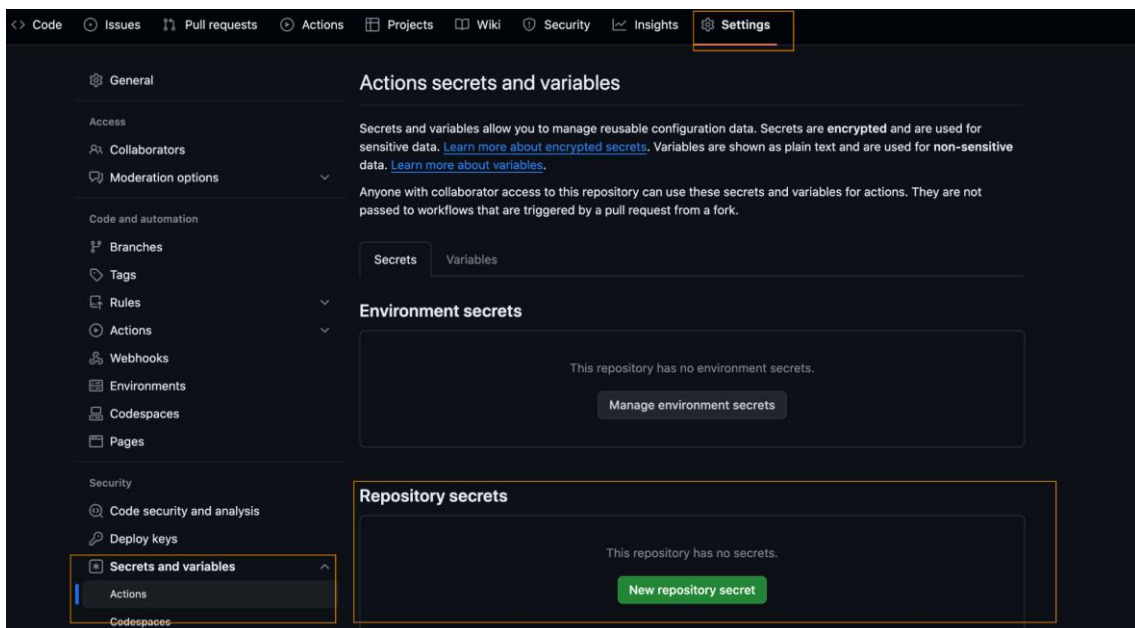
educación IT	Materia: Bootcamp de Devops Engineer	Docente: Zdenko Hraste	Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay		10/12/2023
	Tutor: Facundo Miglio	Comisión: 3	Año: 2023
	DESAFIO N° 16-17		Versión 1

CREACIÓN DEL WORKFLOW: ci.yaml


La automatización de la creación del Dockerfile la realizamos con con GitHub Actions.

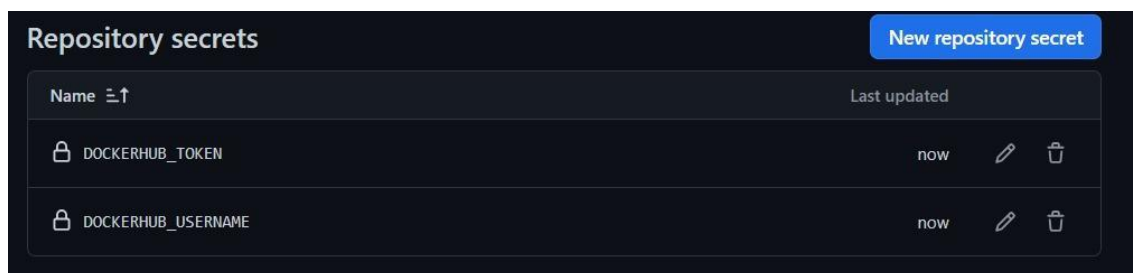


Se almacenan las credenciales de DockerHub dentro del repositorio:



Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1



Se crea el archivo llamado ci.yaml con las configuraciones que se ocuparán de activar el pipeline para ejecutar el docker build y subir esa imagen a DockerHub.

Se definen los pasos a ejecutar:

Workflow trigger: Se activa el workflow cuando se efectúa un push en la rama main. Y si hay cambios en dentro del directorio de Docker.

Jobs 'build': se ejecuta dentro del entorno de Ubuntu, versión latest.

Steps:

Checkout: Descarga el repositorio para que puedan ejecutarse los pasos siguientes.

Login to Docker Hub: inicia sesión.

Set up Docker Buildx: prepara para el proceso de construcción.


Build and push Docker Image: construye y sube la imagen a Docker Hub.

```

1  name: ci
2
3  on:
4    push:
5      branches:
6        - "main"
7      paths:
8        - "Docker/**"
9
10 jobs:
11   build:
12     runs-on: ubuntu-latest
13
14   steps:
15     - name: Checkout Repository
16       uses: actions/checkout@v2
17
18     - name: Login to Docker Hub
19       uses: docker/login-action@v1
20       with:
21         username: ${ secrets.DOCKERHUB_USERNAME }
22         password: ${ secrets.DOCKERHUB_TOKEN }
23
24     - name: Set up Docker Buildx
25       uses: docker/setup-buildx-action@v1
26
27     - name: Build and push Docker Image
28       id: build
29       uses: docker/build-push-action@v2
30       with:
31         context: "Docker"
32         push: true
33         tags: ${ secrets.DOCKERHUB_USERNAME }/plex-media-server:${ github.sha }

```

Educación IT

Educación IT				
	Materia: Bootcamp de Devops Engineer	Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023
	Tutor: Facundo Miglio	Comisión: 3	Año: 2023	
	DESAFIO N° 16-17			Versión 1

Kubernetes

Para convertir los contenedores en recursos de kubernetes debemos instalar minikube en nuestro equipo

Una vez instalado lo iniciamos

```
vagrant@ubuntu-focal:~$ minikube start
minikube v1.32.0 on Ubuntu 20.04 (vbox/amd64)
* Automatically selected the docker driver. Other choices: none, ssh

The requested memory allocation of 1971MiB does not leave room for system overhead (total system memory: 1971MiB). You may face stability issues.
Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1971mb'

Using Docker driver with root privileges
Starting control plane node minikube in cluster minikube
Pulling base image ...
Downloading Kubernetes v1.28.3 preload ...
> gcr.io/k8s-minikube/kicbase... 453.90 MiB / 453.90 MiB 100.00% 6.98 Mi
> preloaded-images-k8s-v18-v1... 403.35 MiB / 403.35 MiB 100.00% 4.61 Mi
Creating docker container (CPUs=2, Memory=1971MB) ...
Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
  * Generating certificates and keys ...
  * Booting up control plane ...
  * Configuring RBAC rules ...
  * Configuring bridge CNI (Container Networking Interface) ...
  * Verifying Kubernetes components...
  * Using image gcr.io/k8s-minikube/storage-provisioner:v5
  Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

vagrant@ubuntu-focal:~$ ls
get-docker.sh  kubectl  minikube-linux-amd64
vagrant@ubuntu-focal:~$ cd ..
vagrant@ubuntu-focal:~/home$ cd ..
vagrant@ubuntu-focal:~$ cd vagrant
vagrant@ubuntu-focal:/vagrant$ ls
Dockerfile  Vagrantfile  bootstrap.sh
vagrant@ubuntu-focal:/vagrant$ docker build . -t docker-on-plex
[+] Building 40.0s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 655B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/ubuntu:latest
=> [1/2] FROM docker.io/library/ubuntu:latest@sha256:8eab65df33a6de2844c9aefd19efebdd8767df5e9185a4ab73af936225685bb
=> => resolve docker.io/library/ubuntu:latest@sha256:8eab65df33a6de2844c9aefd19efebdd8767df5e9185a4ab73af936225685bb
=> => sha256:149d67e29f765f4db2aa52161000e99e389544e25a8f43c8c89d4a445a7ca37 424B / 424B
=> => sha256:b6548aebc8639263e9d8abfe48f8ac3b327182a95335b67572f715c588a968e 2.30kB / 2.30kB
=> => sha256:5e8117c8bd28aacad86f7e76dd3b64734d59c1a0a44541d1806cd8fba38c58 29.59MB / 29.59MB
=> => sha256:8eab65df33a6de2844c9aefd19efebdd8767df5e9185a4ab73af936225685bb 1.13kB / 1.13kB
=> => extracting sha256:5e8117c8bd28aacad86f7e76dd3b64734d59c1a0a44541d1806cd8fba38c58
=> [2/3] RUN apt-get update && apt-get install -y curl && apt-get clean && rm -rf /var/lib/apt/lists/*
=> [3/3] RUN curl -LO "https://downloads.plex.tv/plex-media-server-new/1.32.6.7557-1cf77d501/debian/plexmediaserver_1.32.6.7557-1cf77d501_amd64.deb" && dpkg -i plexmediaserver_1.32.6.7557-1cf77d501
=> exporting to image
docker:default
0.0s
0.0s
0.0s
0.0s
2.3s
4.7s
0.0s
0.0s
3.5s
0.0s
1.0s
16.1s
15.2s
1.7s
```


Accedemos al clúster de minikube por ssh

```
REPOSITORY TAG IMAGE ID CREATED SIZE
docker-on-plex latest 47bc9b963885 26 seconds ago 353MB
gcr.io/k8s-minikube/kicbase v0.0.42 dbc648475485 4 weeks ago 1.7GB
vagrant@ubuntu-focal:/vagrant$ docker run -d --name plex-movies -p 32400:32400 docker-on-plex
7dc68dc4b17c68e789ab8ab82185c2e5a3129ae934678d3cab19e7886972846
vagrant@ubuntu-focal:/vagrant$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
7dc68dc4b17c docker-on-plex "/usr/lib/plexmedias..." 15 seconds ago Up 13 seconds 0.0.0.0:32400->32400/tcp, :::32400->32400/tcp
af0d6e45a58b gcr.io/k8s-minikube/kicbase:v0.0.42 "/usr/local/bin/entr..." 8 minutes ago Up 8 minutes 127.0.0.1:32772->32772/tcp, 127.0.0.1:32771->2376/tcp, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32769->8
vagrant@ubuntu-focal:/vagrant$ minikube ssh
docker@minikube:~$
```

Una vez que inicializamos Minikube procederemos a crear los recursos de kubernetes

Como primer paso debemos crear un archivo YAML denominado deployment.yaml dentro del directorio de nuestro equipo, para luego configurarlo con el Deployment.

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

```

Kubernetes > vim deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: docker-no-plex
5    labels:
6      app: docker-on-plex
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: docker-on-plex
12   template:
13     metadata:
14       labels:
15         app: docker-on-plex
16     spec:
17       containers:
18       - name: docker-on-plex
19         image: florpezz/plex_on_docker:2
20         ports:
21         - containerPort: 32400
22

```

Para aplicar la configuración del Deployment al clúster debemos ejecutar el siguiente comando:


`kubectl apply -f deployment.yaml`

```

vagrant@ubuntu-focal:/vagrant$ kubectl apply -f deployment.yaml
deployment.apps/docker-no-plex created
vagrant@ubuntu-focal:/vagrant$

```

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1

Para obtener todos los recursos en el clúster de Kubernetes ejecutamos

kubectl get all

```
vagrant@ubuntu-focal:/vagrant$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/docker-no-plex-6449d84968-qr77v 1/1     Running   0           5m8s

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes                  ClusterIP     10.96.0.1    <none>        443/TCP    65m

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/docker-no-plex      1/1     1             1           5m8s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/docker-no-plex-6449d84968 1         1         1       5m8s
vagrant@ubuntu-focal:/vagrant$
```

Para verificar el pod:

kubectl get pod

```
vagrant@ubuntu-focal:/vagrant$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
docker-no-plex-6449d84968-qr77v    1/1     Running   0           5m53s
vagrant@ubuntu-focal:/vagrant$
```

Para ver los detalles de los deployments y replicasets:


kubectl get deployments -o wide

```
vagrant@ubuntu-focal:/vagrant$ kubectl get deployment -o wide
NAME          READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES                               SELECTOR
docker-no-plex 1/1     1             1           6m29s docker-on-plex florpezz/plex_on_docker:2 app=docker-on-plex
vagrant@ubuntu-focal:/vagrant$
```

y kubectl get rs

```
vagrant@ubuntu-focal:/vagrant$ kubectl get rs
NAME                                DESIRED   CURRENT   READY   AGE
docker-no-plex-6449d84968          1         1         1       7m6s
vagrant@ubuntu-focal:/vagrant$
```

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1

Para ver la información extendida de los pods


```
vagrant@ubuntu-focal:/vagrant$ kubectl describe pod docker-no-plex-6449d84968-qr77v
Name:          docker-no-plex-6449d84968-qr77v
Namespace:     default
Priority:       0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Wed, 06 Dec 2023 04:40:42 +0000
Labels:        app=docker-on-plex
               pod-template-hash=6449d84968
Annotations:   <none>
Status:        Running
IP:            10.244.0.3
IPs:           IP: 10.244.0.3
Controlled By: ReplicaSet/docker-no-plex-6449d84968
Containers:
  docker-on-plex:
    Container ID:  docker://f4b09ad3b9e03c0c08c2c6a70a4f27dfc5e5f7ba18ceb1f76c686bc051e4d33b
    Image:          florpezz/plex_on_docker:2
    Image ID:       docker-pullable://florpezz/plex_on_docker@sha256:76dfe0dc3b43a22aca6ce7ddd073e74576b832061e828f45fb894660ccb3fc0d
    Port:          32400/TCP
    Host Port:     0/TCP
    State:         Running
      Started:     Wed, 06 Dec 2023 04:41:15 +0000
    Ready:         True
    Restart Count: 0
    Environment:   <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-5rx99 (ro)
Conditions:
  Type            Status
  Initialized     True
  Ready           True
  ContainersReady True
  PodScheduled    True
Volumes:
  kube-api-access-5rx99:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:       kube-root-ca.crt
    ConfigMapOptional:   <nil>
    DownwardAPI:         true
  QoS Class:           BestEffort
  Node-Selectors:      <none>
  Tolerations:         node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                     node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason      Age   From          Message
  ----     -
  Normal   Scheduled   10m   default-scheduler   Successfully assigned default/docker-no-plex-6449d84968-qr77v to minikube
  Normal   Pulling     10m   kubelet          Pulling image "florpezz/plex_on_docker:2"
  Normal   Pulled      9m56s kubelet          Successfully pulled image "florpezz/plex_on_docker:2" in 31.748s (31.749s including waiting)
  Normal   Created     9m55s kubelet          Created container docker-on-plex
  Normal   Started     9m55s kubelet          Started container docker-on-plex
vagrant@ubuntu-focal:/vagrant$
```

Para ver la información sobre los nodos del clúster:

kubectl get nodes

```
vagrant@ubuntu-focal:/vagrant$ kubectl get nodes
NAME          STATUS    ROLES          AGE   VERSION
minikube      Ready     control-plane   72m   v1.28.3
```

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1


información adicional

minikube addons list

```
vagrant@ubuntu-focal:~$ minikube addons list
```

ADDON NAME	PROFILE	STATUS	MAINTAINER
ambassador	minikube	disabled	3rd party (Ambassador)
auto-pause	minikube	disabled	minikube
cloud-spanner	minikube	disabled	Google
csi-hostpath-driver	minikube	disabled	Kubernetes
dashboard	minikube	disabled	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)
helm-tiller	minikube	disabled	3rd party (Helm)
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
kubevirt	minikube	disabled	3rd party (KubeVirt)
logviewer	minikube	disabled	3rd party (unknown)
metallb	minikube	disabled	3rd party (MetalLB)
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)
volumesnapshots	minikube	disabled	Kubernetes

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

Helm y ArgoCD

Para convertir los recursos de kubernetes en un paquete Helm, primero debemos instalar la aplicación en nuestro entorno y luego realizar las siguientes configuraciones

Instalación de Helm


```
vagrant@ubuntu-focal:/vagrant$ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
vagrant@ubuntu-focal:/vagrant$ chmod 700 get_helm.sh
vagrant@ubuntu-focal:/vagrant$ ./get_helm.sh
Downloading https://get.helm.sh/helm-v3.13.1-linux-amd64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/helm
vagrant@ubuntu-focal:/vagrant$
```

```
vagrant@ubuntu-focal:~$ sudo snap install helm --classic
helm 3.10.1 from Snapcrafters installed
vagrant@ubuntu-focal:~$
```

Creemos la raíz donde luego vamos a configurar el paquete Helm.

```
vagrant@ubuntu-focal:/vagrant$ mkdir charts
vagrant@ubuntu-focal:/vagrant$ cd charts
vagrant@ubuntu-focal:/vagrant/charts$ helm create desafio16-17
Creating desafio16-17
vagrant@ubuntu-focal:/vagrant/charts$ ls
desafio16-17
vagrant@ubuntu-focal:/vagrant/charts$ cd desafio16-17/
vagrant@ubuntu-focal:/vagrant/charts/desafio16-17$ ls
Chart.yaml  charts  templates  values.yaml
vagrant@ubuntu-focal:/vagrant/charts/desafio16-17$
```

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

Verificamos el estado de la creación

```
vagrant@ubuntu-focal:~$ helm status docker-on-plex-4
NAME: docker-on-plex-4
LAST DEPLOYED: Sun Dec 10 14:34:54 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=desafio16-17,app.kubernetes.io/instance=docker-on-plex-4" -o jsonpath="{.items[0].metadata.name}")
  export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
vagrant@ubuntu-focal:~$ helm status docker-on-plex-4 --show-resources
NAME: docker-on-plex-4
LAST DEPLOYED: Sun Dec 10 14:34:54 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
RESOURCES:
==> v1/ServiceAccount
NAME                SECRETS  AGE
docker-on-plex-4-desafio16-17  0        81s

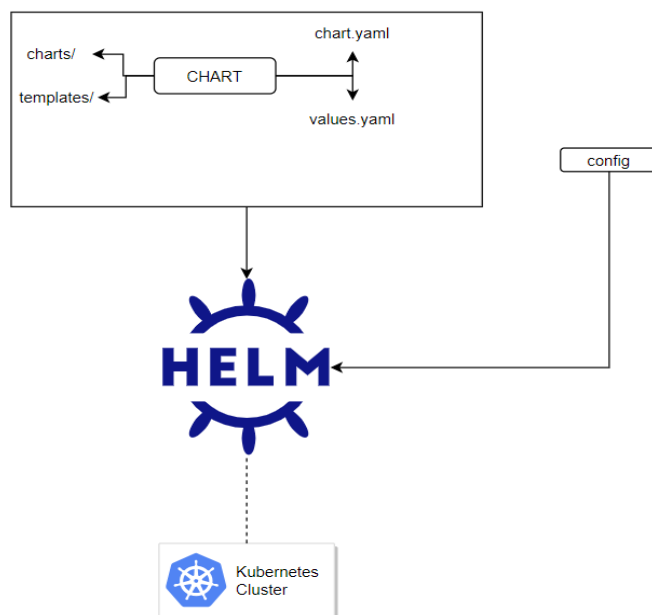
==> v1/Service
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)  AGE
docker-on-plex-4-desafio16-17  ClusterIP   10.109.106.191  <none>       80/TCP   81s

==> v1/Deployment
NAME                READY  UP-TO-DATE  AVAILABLE  AGE
docker-on-plex-2    1/1    1            1          81s


==> v1/Pod(related)
NAME                READY  STATUS   RESTARTS  AGE
docker-on-plex-2-6449d84968-4d42k  1/1    Running   0          81s
docker-on-plex-2-6449d84968-j57nz  1/1    Running   0          27m

NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=desafio16-17,app.kubernetes.io/instance=docker-on-plex-4" -o jsonpath="{.items[0].metadata.name}")
  export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
vagrant@ubuntu-focal:~$
```

Estructura del Chart



Educación IT


Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

Para nuestro trabajo vamos a editar los archivos values.yaml y deployment.yaml que se encuentra dentro de la carpeta /templates

values.yaml

```
charts > desafio16-17 > ! values.yaml
1  # Default values for desafio16-17.
2  # This is a YAML-formatted file.
3  # Declare variables to be passed into your templates.
4  # Default values for desafio16-17.
5  # This is a YAML-formatted file.
6  # Declare variables to be passed into your templates.
7  #inicio de valores personalizados
8  metadata:
9    | name: docker-no-plex
10 # labels:
11 #   app: docker-on-plex
12 #spec:
13 # replicas: 1
14 # selector:
15 #   matchLabels:
16 #     app: docker-on-plex
17 # template:
18 #   metadata:
19 #     labels:
20 #       app: docker-on-plex
21 #   spec:
22 #     containers:
23 # - name: docker-on-plex
24 #   image: florpezz/plex_on_docker:2
25 #   ports:
26 #     - containerPort: 32400
```


Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1


deployment.yaml

```
charts > desafio16-17 > templates > ! deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: {{ .Values.metadata.name }}
5    labels:
6      app: docker-on-plex
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: docker-on-plex
12   template:
13     metadata:
14       labels:
15         app: docker-on-plex
16     spec:
17       containers:
18         - name: docker-on-plex
19           image: florpezz/plex_on_docker:2
20           ports:
21             - containerPort: 32400
```

Luego creamos el chart de la aplicación

```
vagrant@ubuntu-focal:~/vagrant/charts/desafio16-17$ helm install docker-on-plex-4 .
NAME: docker-on-plex-4
LAST DEPLOYED: Sun Dec 10 14:34:54 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=desafio16-17,app.kubernetes.io/instance=docker-on-plex-4" -o jsonpath="{.items[0].metadata.name}")
  export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
vagrant@ubuntu-focal:~/vagrant/charts/desafio16-17$
```


Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1

También debemos abrir los puertos para poder ingresar

```

vagrant@ubuntu-focal:~$ vagrant ssh -- -L 8080:localhost:8080
PS C:\Users\Alexis Roldan\Desktop\desafio16-17> cd .\Vagrant\
PS C:\Users\Alexis Roldan\Desktop\desafio16-17\Vagrant> vagrant ssh -- -L 8080:localhost:8080
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-163-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information disabled due to load higher than 6.0

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

53 updates can be applied immediately.
38 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

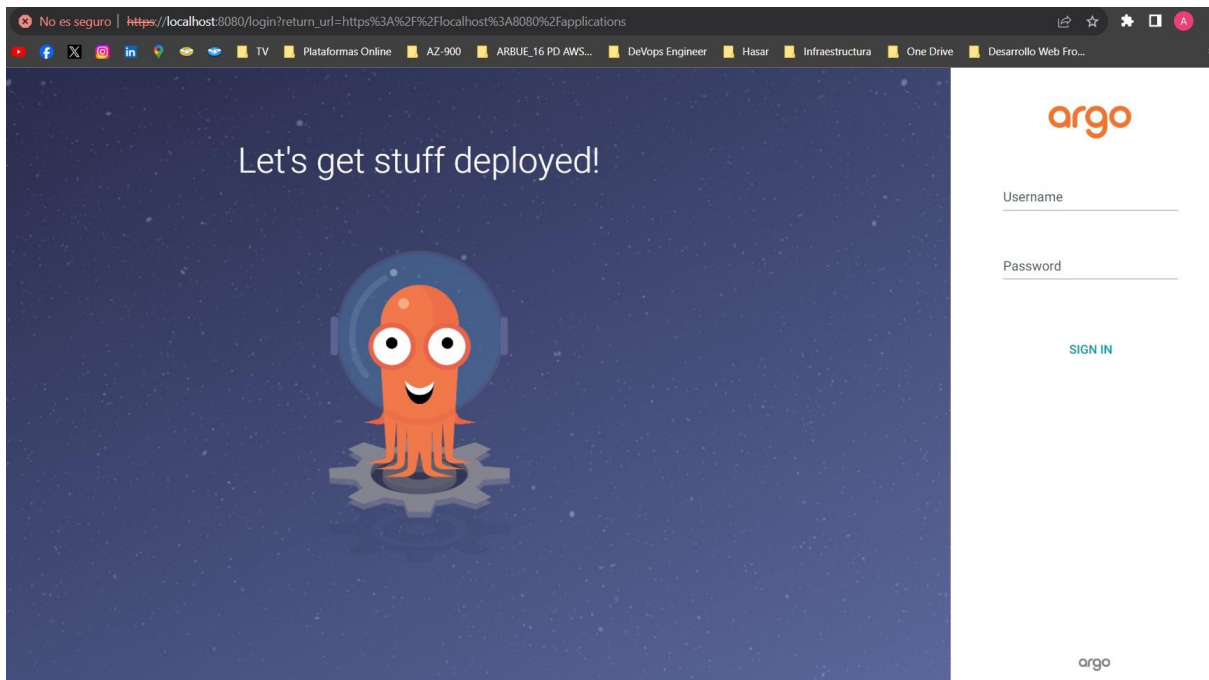
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Dec 10 13:52:06 2023 from 10.0.2.2
vagrant@ubuntu-focal:~$
  
```

En el navegador vamos a poder ingresar con la siguiente url: <https://localhost:8080> a la interfaz de ArgoCD

Educación IT

Educación IT				
	Materia: Bootcamp de Devops Engineer	Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023
	Tutor: Facundo Miglio	Comisión: 3	Año: 2023	
	DESAFIO N° 16-17			Versión 1

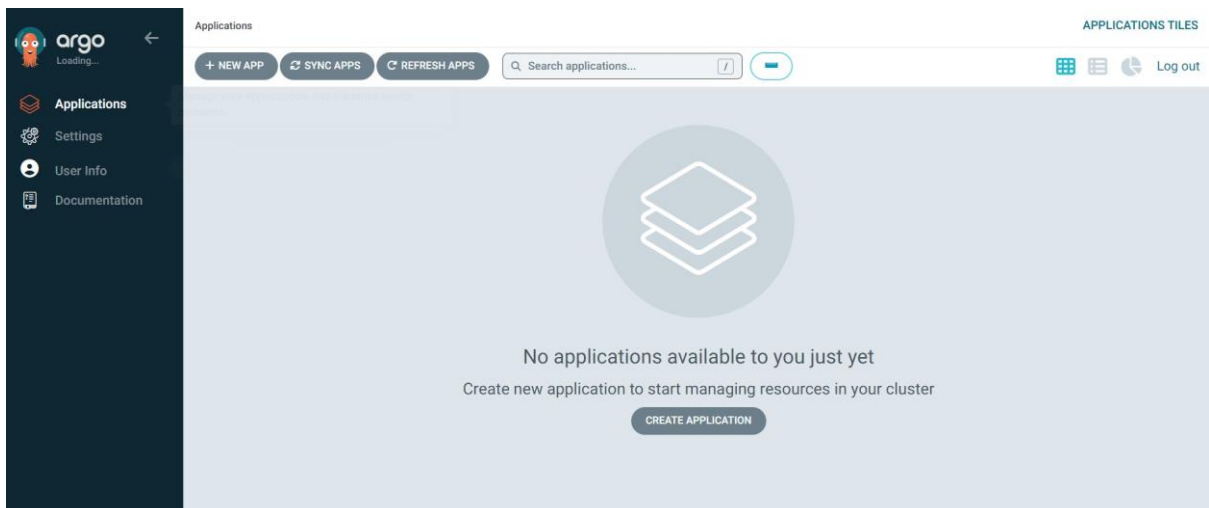


Para ingresar debemos colocar la Password que se encuentra ejecutando este comando


```
vagrant@ubuntu-focal:~$ argocd admin initial-password -n argocd
jFAcqTwkNOzKQjan

This password must be only used for first time login. We strongly recommend you update the password using `argocd account update-password`.
vagrant@ubuntu-focal:~$
```

Ingresando las credenciales vamos a poder ingresar a la interfaz



Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1


también para una mejor administración debemos ingresar por terminal

```
vagrant@ubuntu-focal:~$ argocd login localhost:8080
WARNING: server certificate had error: tls: failed to verify certificate: x509: certificate signed by unknown authority. Proceed insecurely (y/n)? y
Username: admin
Password:
'admin:login' logged in successfully
Context 'localhost:8080' updated
vagrant@ubuntu-focal:~$
```

Luego agregamos el clúster de minikube

```
vagrant@ubuntu-focal:~$ kubectl config get-contexts -o name
minikube
vagrant@ubuntu-focal:~$ argocd cluster add minikube
WARNING: This will create a service account 'argocd-manager' on the cluster referenced by context 'minikube' with full cluster level privileges. Do you want to continue [y/N]? y
INFO[0004] ServiceAccount "argocd-manager" created in namespace "kube-system"
INFO[0004] ClusterRole "argocd-manager-role" created
INFO[0004] ClusterRoleBinding "argocd-manager-role-binding" created
INFO[0009] Created bearer token secret for ServiceAccount "argocd-manager"
Cluster "https://192.168.49.2:8443" added
```

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

A continuación crearemos la aplicación en la interfaz de ArgoCD

CREATE

CANCEL

GENERAL

Application Name

docker-on-plex

Project Name

default

SYNC POLICY

Automatic

☐ PRUNE RESOURCES
 ☐ SELF HEAL
 ☐ SET DELETION FINALIZER

SYNC OPTIONS

☐ SKIP SCHEMA VALIDATION
 ☐ PRUNE LAST
 ☐ RESPECT IGNORE DIFFERENCES

☐ AUTO-CREATE NAMESPACE
 ☐ APPLY OUT OF SYNC ONLY
 ☐ SERVER-SIDE APPLY

PRUNE PROPAGATION POLICY:

foreground

☐ REPLACE
 ☐ RETRY

Repository URL

https://github.com/alexlpda1420/desafio16-17.git

GIT

Revision

HEAD

Branches

Path

Helm/charts/desafio16-17

DESTINATION

Cluster URL


https://kubernetes.default.svc

URL



Namespace

Desafio16-17

Educación IT

Educación IT				
	Materia: Bootcamp de Devops Engineer	Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023
	Tutor: Facundo Miglio	Comisión: 3	Año: 2023	
	DESAFIO N° 16-17			Versión 1

Una vez que creamos la aplicación nos mostrara un detalle de la aplicación creada pero aún no sincronizada




 **docker-on-plex** 

Project:

default

Labels:

Status:

 Missing  OutOfSync  Syncing

Repository:

<https://github.com/alexlpda1420/desafio1...>

Target Re...

HEAD

Path:

Helm/charts/desafio16-17

Destinati...

in-cluster

Namespa...


Desafio16-17


Created At:


12/10/2023 19:21:31 (a few seconds ago)




Last Sync:

12/10/2023 19:21:33 (a few seconds ago)


 SYNC

 REFRESH

 DELETE

SUMMARY	PARAMETERS	MANIFEST	 DIFF	EVENTS
DOCKER-ON-PLEX				
PROJECT	default			
ANNOTATIONS				
CLUSTER	in-cluster (https://kubernetes.default.svc)			
NAMESPACE	Desafio16-17 			
CREATED AT	12/10/2023 19:21:31 (a minute ago)			
REPO URL	https://github.com/alexlpda1420/desafio16-17.git			
TARGET REVISION	HEAD			
PATH	Helm/charts/desafio16-17			
SYNC OPTIONS				
RETRY OPTIONS	Retry disabled			
STATUS	 OutOfSync from HEAD (27cdd0c)			

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay			10/12/2023	
	Tutor: Facundo Miglio		Comisión: 3		Año: 2023
	DESAFIO N° 16-17				Versión 1

Applications / Q docker-on-plex

DETAILS DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

APP HEALTH Missing

SYNC STATUS OutOfSync from HEAD (27cdd0c)

Auto sync is enabled.
Author: Alexis Roldan <a.eroldan@hotmail.com>
Comment: Acomodando archivos

LAST SYNC Syncing

Running a few seconds ago (Sun Dec 10 2023 19:22:07 GMT-0600)
one or more objects failed to apply, reason: namespaces "Desafio16-17" not found.
Retrying attempt #3 at 1:22AM.

Para que termine de sincronizar, debemos ver el estado por terminal ejecutando el comando


`argocd app get docker-on-plex-2`


```
vagrant@ubuntu-focal:~$ argocd app get docker-on-plex-2
Name:          argocd/docker-on-plex-2
Project:       default
Server:        https://kubernetes.default.svc
Namespace:     docker-on-plex-2
URL:           https://localhost:8080/applications/docker-on-plex-2
Repo:          https://github.com/alexlpda1420/desafio16-17.git
Target:        HEAD
Path:          Helm/charts/desafio16-17
Syncwindow:    Sync Allowed
Sync Policy:   Automated
Sync Status:   OutOfSync from HEAD (27cdd0c)
Health Status: Missing

GROUP KIND      NAMESPACE      NAME                                STATUS  HEALTH  HOOK  MESSAGE
-----
ServiceAccount docker-on-plex-2 docker-on-plex-2-desafio16-17 OutOfSync Missing namespaces "docker-on-plex-2" not found
Service        docker-on-plex-2 docker-on-plex-2-desafio16-17 OutOfSync Missing namespaces "docker-on-plex-2" not found
apps Deployment docker-on-plex-2 docker-on-plex-2 OutOfSync Missing namespaces "docker-on-plex-2" not found
```

En nuestro caso nos arrojaba un error porque el namespace declarado en la creación de la aplicación no existía, se corrigió y se colocó el namespace “default” que si existe y sincronizo sin inconvenientes nuestra aplicación

Educación IT

Educación IT					
	Materia: Bootcamp de Devops Engineer		Docente: Zdenko Hraste		Fecha
	Alumno: Alexis Roldan – Florencia Pezzella – Mauro Usay				10/12/2023
	Tutor: Facundo Miglio		Comisión: 3	Año: 2023	
	DESAFIO N° 16-17				Versión 1


docker-on-plex-2
☆

Project: default

Labels:

Status: ♥ Healthy ✓ Synced

Repository: <https://github.com/alexlpda1420/desafio1...>

Target Re... HEAD

Path: Helm/charts/desafio16-17

Destinati... in-cluster

Namespa... default

Created At: 12/10/2023 19:52:37 (a few seconds ago)

Last Sync: 12/10/2023 19:52:38 (a few seconds ago)

↻ SYNC
↻ REFRESH
✕ DELETE

Entregable

- Instructivo= Desafio - Módulo 16-17.pdf
- Link al repositorio de código url: <https://github.com/alexlpda1420/desafio16-17>
- Link al repositorio de la imagen url: https://hub.docker.com/r/florpezz/plex_on_docker
- Una carpeta con una copia del repositorio de código: carpeta contenedora "Modulo 16-17"