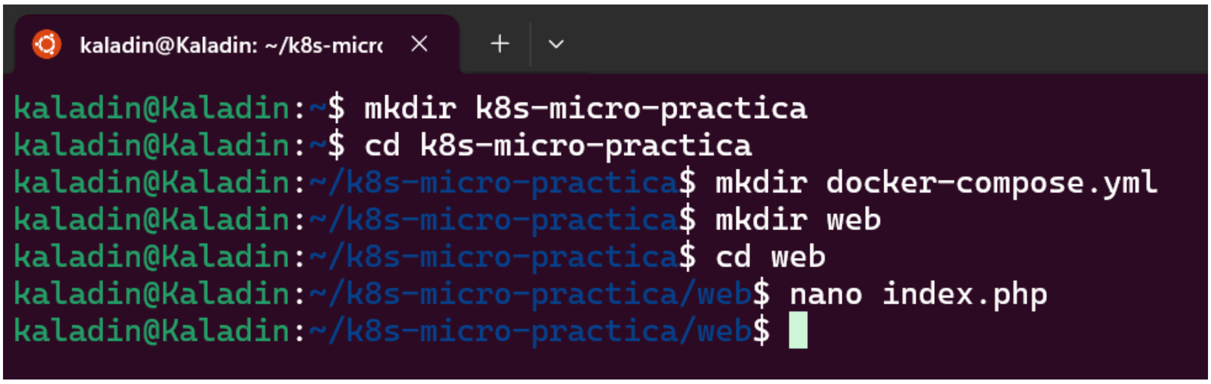
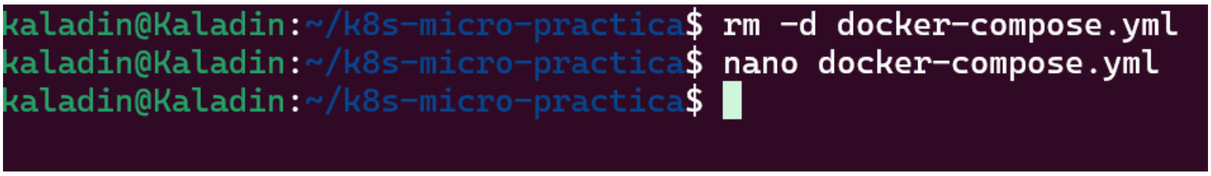
PARTE 1



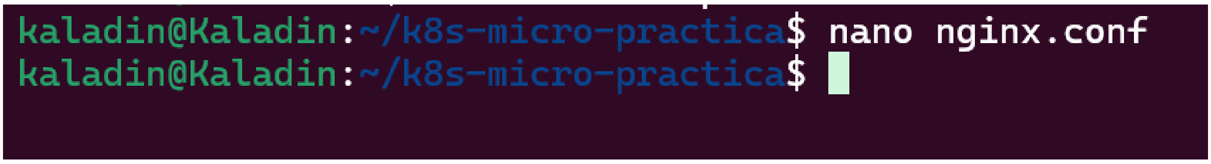
```
kaladin@Kaladin:~$ mkdir k8s-micro-practica
kaladin@Kaladin:~$ cd k8s-micro-practica
kaladin@Kaladin:~/k8s-micro-practica$ mkdir docker-compose.yml
kaladin@Kaladin:~/k8s-micro-practica$ mkdir web
kaladin@Kaladin:~/k8s-micro-practica$ cd web
kaladin@Kaladin:~/k8s-micro-practica/web$ nano index.php
kaladin@Kaladin:~/k8s-micro-practica/web$
```

Creo la estructura del proyecto.



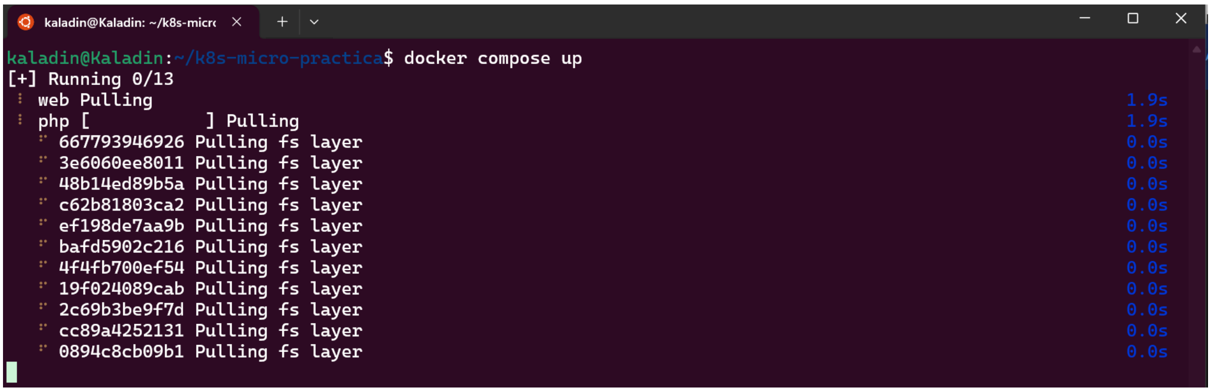
```
kaladin@Kaladin:~/k8s-micro-practica$ rm -d docker-compose.yml
kaladin@Kaladin:~/k8s-micro-practica$ nano docker-compose.yml
kaladin@Kaladin:~/k8s-micro-practica$
```

Creo el docker compose.



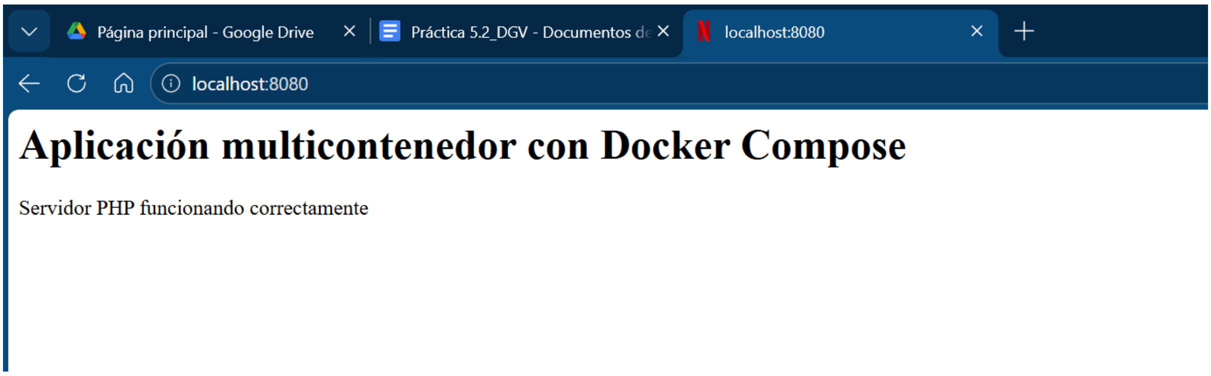
```
kaladin@Kaladin:~/k8s-micro-practica$ nano nginx.conf
kaladin@Kaladin:~/k8s-micro-practica$
```

Creo el nginx.



```
kaladin@Kaladin:~/k8s-micro-practica$ docker compose up
[+] Running 0/13
 ⠸ web Pulling                                                        1.9s
 ⠸ php [          ] Pulling                                           1.9s
   ⠋ 667793946926 Pulling fs layer                                    0.0s
   ⠋ 3e6060ee8011 Pulling fs layer                                    0.0s
   ⠋ 48b14ed89b5a Pulling fs layer                                    0.0s
   ⠋ c62b81803ca2 Pulling fs layer                                    0.0s
   ⠋ ef198de7aa9b Pulling fs layer                                    0.0s
   ⠋ bafd5902c216 Pulling fs layer                                    0.0s
   ⠋ 4f4fb700ef54 Pulling fs layer                                    0.0s
   ⠋ 19f024089cab Pulling fs layer                                    0.0s
   ⠋ 2c69b3be9f7d Pulling fs layer                                    0.0s
   ⠋ cc89a4252131 Pulling fs layer                                    0.0s
   ⠋ 0894c8cb09b1 Pulling fs layer                                    0.0s
```

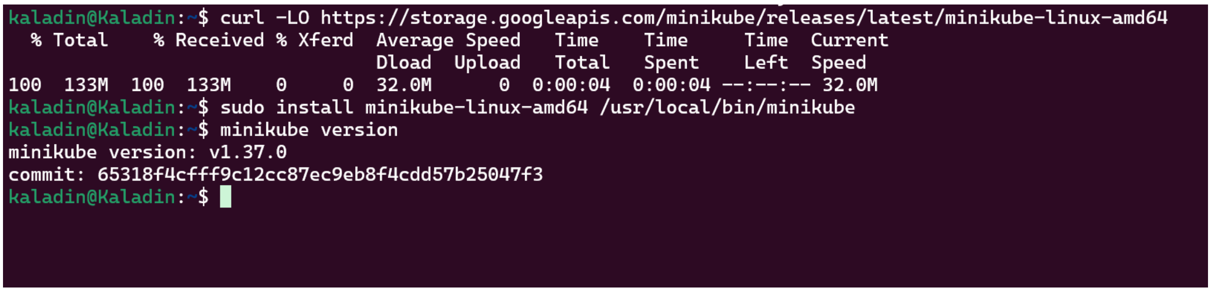
Arranco la aplicación.


Página principal - Google Drive
Práctica 5.2_DGV - Documentos de
localhost:8080
localhost:8080

Aplicación multicontenedor con Docker Compose

Servidor PHP funcionando correctamente

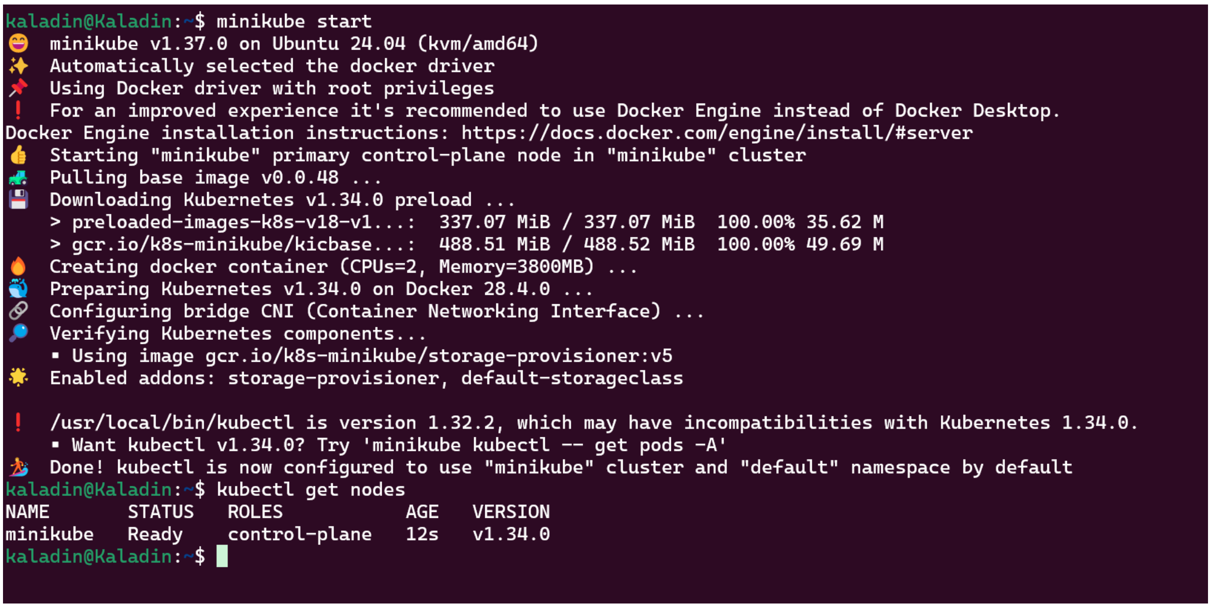
PARTE 2



```
kaladin@Kaladin:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  133M  100  133M    0     0  32.0M      0  0:00:04  0:00:04 --:--:-- 32.0M
kaladin@Kaladin:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
kaladin@Kaladin:~$ minikube version
minikube version: v1.37.0
commit: 65318f4cfff9c12cc87ec9eb8f4cdd57b25047f3
kaladin@Kaladin:~$ 
```

Instalo Minikube.

PARTE 3

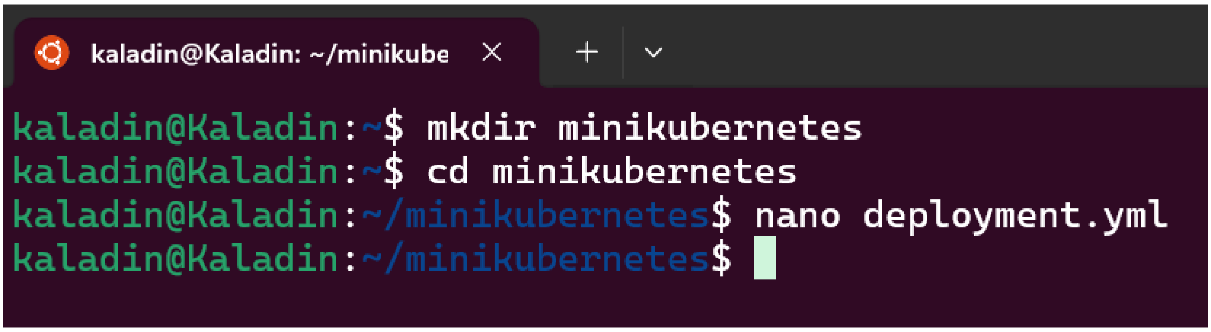


```
kaladin@Kaladin:~$ minikube start
😄  minikube v1.37.0 on Ubuntu 24.04 (kvm/amd64)
✨  Automatically selected the docker driver
📌  Using Docker driver with root privileges
❗  For an improved experience it's recommended to use Docker Engine instead of Docker Desktop.
Docker Engine installation instructions: https://docs.docker.com/engine/install/#server
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.48 ...
💾  Downloading Kubernetes v1.34.0 preload ...
    > preloaded-images-k8s-v18-v1...:  337.07 MiB / 337.07 MiB  100.00% 35.62 M
    > gcr.io/k8s-minikube/kicbase...:  488.51 MiB / 488.52 MiB  100.00% 49.69 M
🔥  Creating docker container (CPUs=2, Memory=3800MB) ...
🐳  Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass

❗  /usr/local/bin/kubectl is version 1.32.2, which may have incompatibilities with Kubernetes 1.34.0.
    ▪ Want kubectl v1.34.0? Try 'minikube kubectl -- get pods -A'
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
kaladin@Kaladin:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   12s   v1.34.0
kaladin@Kaladin:~$ 
```

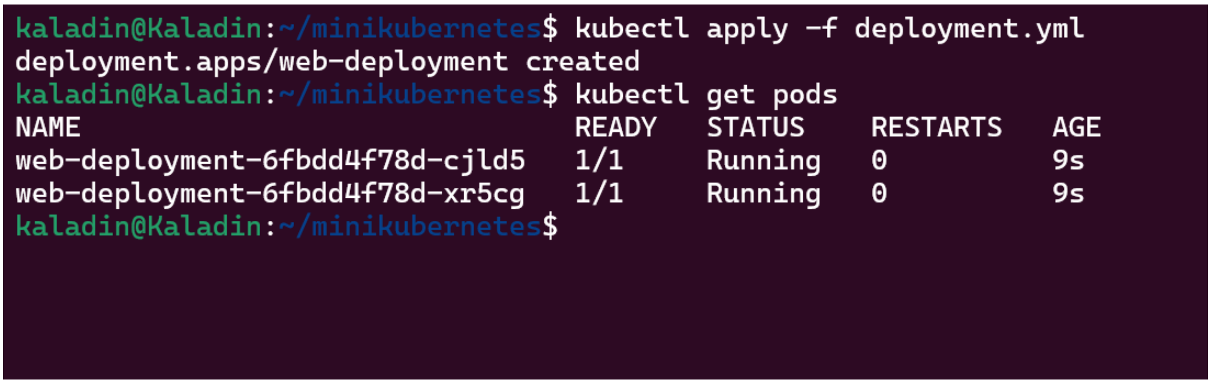
Arranco Kubernetes.

PARTE 4



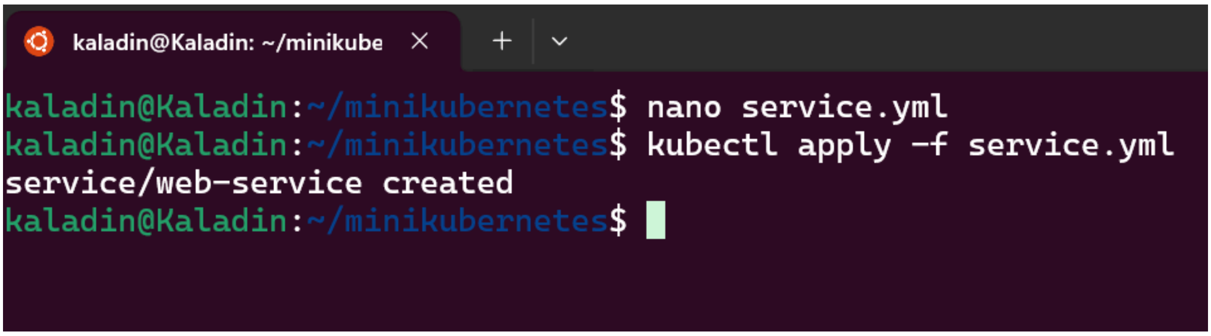
```
kaladin@Kaladin:~$ mkdir minikubernetes
kaladin@Kaladin:~$ cd minikubernetes
kaladin@Kaladin:~/minikubernetes$ nano deployment.yml
kaladin@Kaladin:~/minikubernetes$
```

Creo el yml.



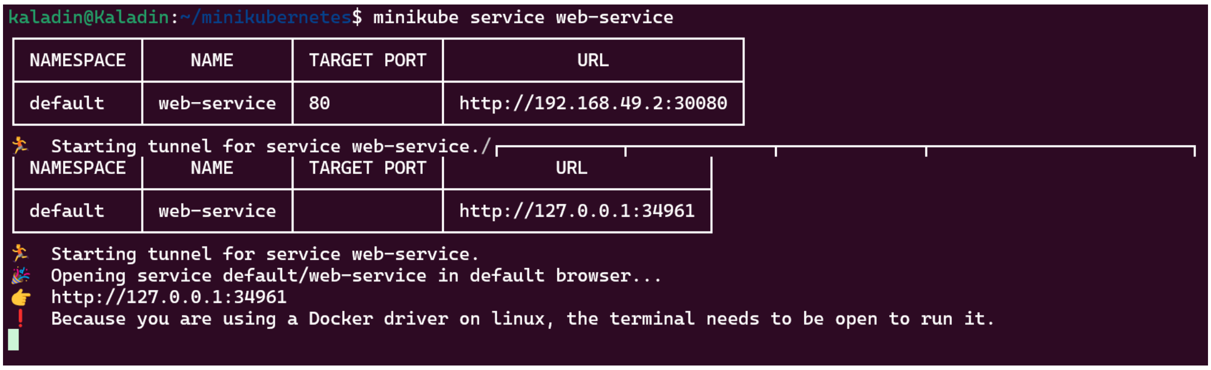
```
kaladin@Kaladin:~/minikubernetes$ kubectl apply -f deployment.yml
deployment.apps/web-deployment created
kaladin@Kaladin:~/minikubernetes$ kubectl get pods
NAME                              READY   STATUS    RESTARTS   AGE
web-deployment-6fbdd4f78d-cjld5   1/1     Running   0          9s
web-deployment-6fbdd4f78d-xr5cg   1/1     Running   0          9s
kaladin@Kaladin:~/minikubernetes$
```

Aplico el deployment.

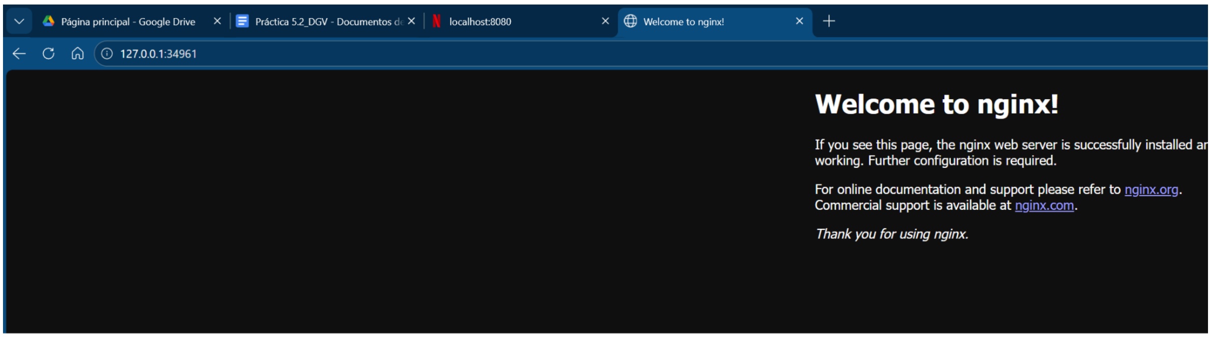


```
kaladin@Kaladin:~/minikubernetes$ nano service.yml
kaladin@Kaladin:~/minikubernetes$ kubectl apply -f service.yml
service/web-service created
kaladin@Kaladin:~/minikubernetes$
```

Creo el yml de service y lo aplico.



```
kaladin@Kaladin:~/minikubernetes$ minikube service web-service
┌───────────┬─────────────┬─────────────┬───────────────────────────┐
│ NAMESPACE │    NAME     │ TARGET PORT │            URL            │
├───────────┼─────────────┼─────────────┼───────────────────────────┤
│ default   │ web-service │ 80          │ http://192.168.49.2:30080 │
└───────────┴─────────────┴─────────────┴───────────────────────────┘
🏃  Starting tunnel for service web-service./┌───────────┬─────────────┬─────────────┬───────────────────────────┐
│ NAMESPACE │    NAME     │ TARGET PORT │          URL           │
├───────────┼─────────────┼─────────────┼────────────────────────┤
│ default   │ web-service │             │ http://127.0.0.1:34961 │
└───────────┴─────────────┴─────────────┴────────────────────────┘
🏃  Starting tunnel for service web-service.
🎉  Opening service default/web-service in default browser...
👉  http://127.0.0.1:34961
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```


Página principal - Google Drive
Práctica 5.2_DGV - Documentos d
localhost:8080
Welcome to nginx!
127.0.0.1:34961

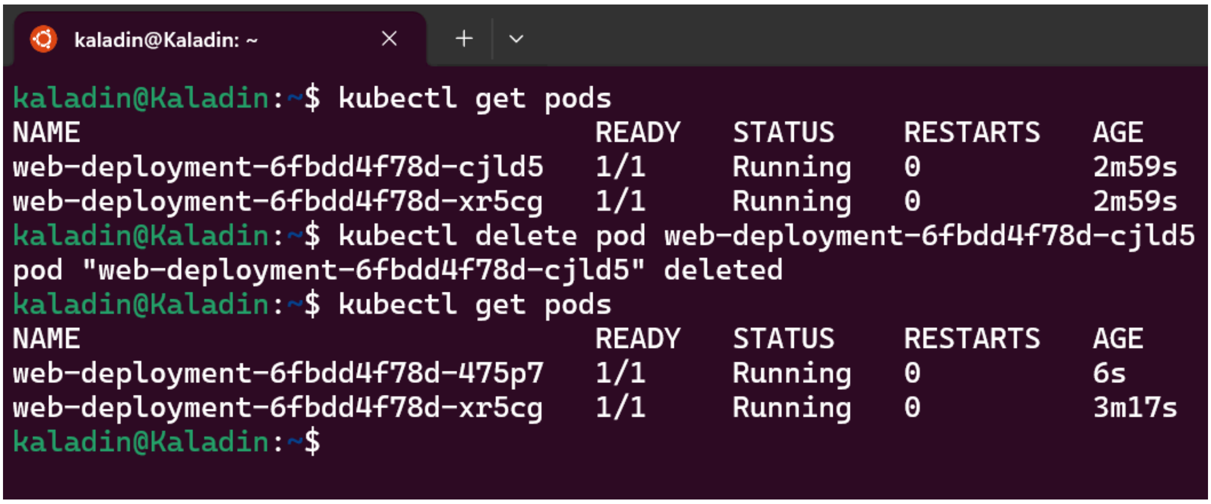
Welcome to nginx!

If you see this page, the nginx web server is successfully installed ar working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

PARTE 5



```
kaladin@Kaladin:~$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
web-deployment-6fbdd4f78d-cjld5    1/1     Running   0          2m59s
web-deployment-6fbdd4f78d-xr5cg    1/1     Running   0          2m59s
kaladin@Kaladin:~$ kubectl delete pod web-deployment-6fbdd4f78d-cjld5
pod "web-deployment-6fbdd4f78d-cjld5" deleted
kaladin@Kaladin:~$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
web-deployment-6fbdd4f78d-475p7    1/1     Running   0          6s
web-deployment-6fbdd4f78d-xr5cg    1/1     Running   0          3m17s
kaladin@Kaladin:~$
```

CUESTIONES

1. Un pod es un contenedor de Kubernetes que ejecuta una app.
2. Cuando se elimina un pod, se crea automáticamente uno nuevo.
3. Kubernetes sirve para mantener contenedores en más de una máquina, al contrario que con Docker.

TABLA

ASPECTO	DOCKER COMPOSE	KUBERNETES
num contenedores	Varios	Muchos
num servidores	1	Varios
escalado	Manual	Automático
tolerancia a fallos	Baja (hay que reiniciar el contenedor)	Alta (pods renovados automáticamente)

CONCLUSIÓN

Docker es útil en entornos en los que posees una sola máquina y necesitas tener varios contenedores, hacer pruebas varias de manera sencilla, etc. Luego con Kubenetes puedes sacarlo a producción para que sea más tolerante a los fallos.