

1.Preparar el entorno.

The image shows a Windows terminal window and the Docker Desktop application. The terminal window displays the output of the `kubectl get nodes` command, showing a single node named `docker-desktop` with status `Ready` and role `control-plane`. The Docker Desktop application is open to the **Kubernetes** settings page. The **Enable Kubernetes** toggle is turned on. The **Cluster** section shows a single cluster named `docker-desktop` with the `kubeadm, single-node` provisioning method, which is currently **Running**. The **Cluster settings** section is visible below. The bottom status bar of Docker Desktop indicates that both the **Engine** and **Kubernetes** are running, with resource usage for RAM, CPU, and Disk.

```
Microsoft Windows [Versión 10.0.26100.4351]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\GamerPc>kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
docker-desktop Ready    control-plane 5m34s v1.32.2

C:\Users\GamerPc>
```

Kubernetes

Enable Kubernetes
Start a Kubernetes single or multi-node cluster when starting Docker Desktop.

Cluster

Running
Started 5 minutes ago
Reset Cluster

Cluster settings
Choose cluster provisioning method

Cancel Apply & restart

Engine running | Kubernetes running RAM 3.86 GB CPU 3.39% Disk: 3.39 GB used (limit 1006.85 GB)

2.Desplegar la primera aplicación

The image shows a web browser window displaying the **Welcome to nginx!** page, indicating that the nginx web server is successfully installed and working. Below the browser window, a Windows terminal window shows the commands used to create and expose the nginx deployment. The terminal output includes the `kubectl get nodes` command, the `kubectl create deployment` command, the `kubectl get pods` command, the `kubectl expose deployment` command, and the `kubectl get services` command. The output of the `kubectl get pods` command shows a single pod named `nginx-deploy-c9d9f6c6c-7vjg5` with status `ContainerCreating`. The output of the `kubectl get services` command shows a single service named `nginx-deploy` with type `NodePort` and port `80`.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and 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.

```
Microsoft Windows [Versión 10.0.26100.4351]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\GamerPc>kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
docker-desktop Ready    control-plane 5m34s v1.32.2

C:\Users\GamerPc>kubectl create deployment nginx-deploy --image=nginx
deployment.apps/nginx-deploy created

C:\Users\GamerPc>kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-deploy-c9d9f6c6c-7vjg5  0/1     ContainerCreating  0          6s

C:\Users\GamerPc>kubectl expose deployment nginx-deploy --type=NodePort --port=80
service/nginx-deploy exposed

C:\Users\GamerPc>kubectl get services
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes    ClusterIP   10.96.0.1     <none>         443/TCP          10m
nginx-deploy   NodePort    10.111.57.210 <none>         80:31175/TCP     7s

C:\Users\GamerPc>
```

3.Escalamos la aplicación

```
C:\Users\GamerPc>kubectl scale deployment nginx-deploy --replicas=3
deployment.apps/nginx-deploy scaled

C:\Users\GamerPc>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deploy-c9d9f6c6c-7vjg5       1/1     Running   0           3m20s
nginx-deploy-c9d9f6c6c-jctzh       1/1     Running   0           7s
nginx-deploy-c9d9f6c6c-nqrsz       1/1     Running   0           7s

C:\Users\GamerPc>|
```

4.Elimanos todo.

```
C:\Users\GamerPc>kubectl delete service nginx-deploy
service "nginx-deploy" deleted

C:\Users\GamerPc>kubectl delete deployment nginx-deploy
deployment.apps "nginx-deploy" deleted
```