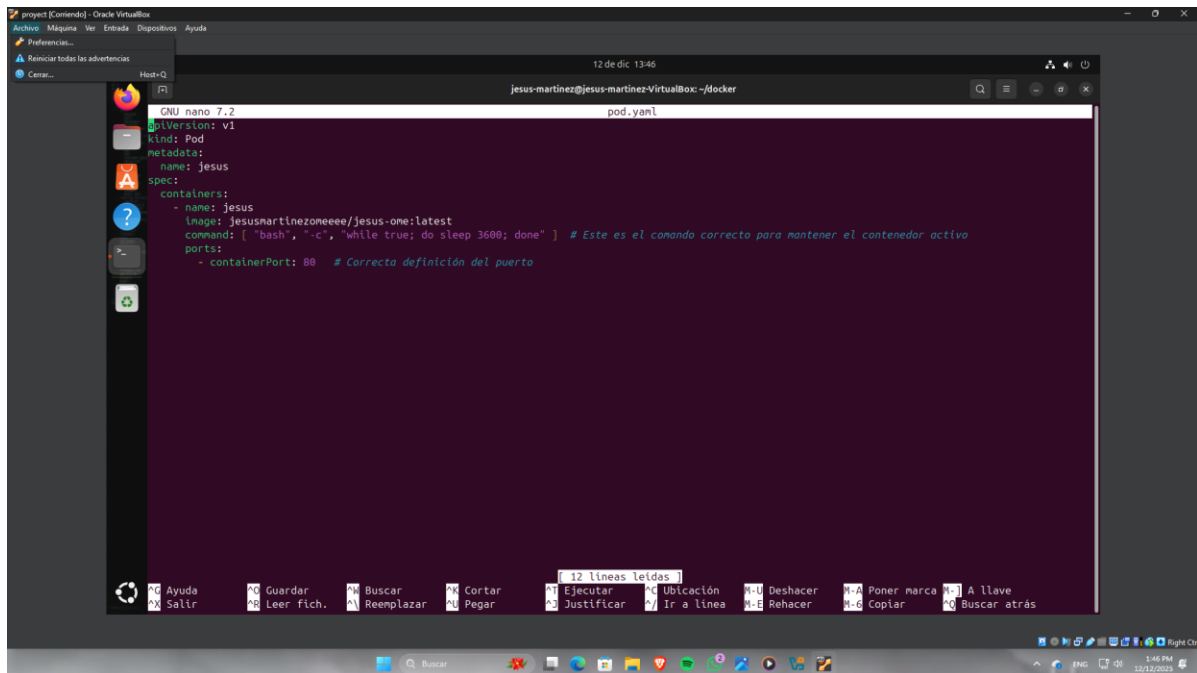


## 6.2 CREACION Y DESPLIEGUE DEL POD

1. creamos un archivo .YAML donde pondremos las instrucciones para que el kubernetes cree el pod, colocamos primero la version API de kubernetes, indicamos el tipo, le damos un nombre al pod, y despues seguimos con las especificaciones, un pod puede contener uno o muchos contenedores, despues el nombre del contenedor, el nombre de la imagen docker, y en command ponemos un comando para que el contenedor se mantenga activo en un bucle y por ultimo el puerto



```
GNU nano 7.2 pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: jesus
spec:
  containers:
  - name: jesus
    image: jesusmartinez000000/jesus-one:latest
    command: [ "bash", "-c", "while true; do sleep 3600; done" ] # Este es el comando correcto para mantener el contenedor activo
    ports:
    - containerPort: 80 # Correcta definici3n del puerto
```

2. descargamos la imagen a trabajar, despues creamos el pod apartir del archivo yaml que hicimos (si ya existe el pod, entonces este se actualiza), despues en una lista revisamos los pods activos, con describe podemos ver toda la informacion interna del pod, desde su ip, el nombre la informacion del contenedor, si presenta fallas o reinicios.

```

jesus-martinez@jesus-martinez-VirtualBox: ~/docker
Using default tag: latest
latest: Pulling from jesusmartinezzone/jesus-one
Digest: sha256:3723ad30e6dae3963640e9d1b0390851aa6e1a1dce4dd6103889717da95108e4
Status: Downloaded newer image for jesusmartinezzone/jesus-one:latest
docker.io/jesusmartinezzone/jesus-one:latest
jesus-martinez@jesus-martinez-VirtualBox: ~/docker$ kubectl apply -f pod.yaml
pod/jesus unchanged
jesus-martinez@jesus-martinez-VirtualBox: ~/docker$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
jesus     1/1     Running   2 (3m4s ago)  13h
nginx-deployment-556dcfc9-9pkp5  1/1     Running   2 (3m4s ago)  13h
nginx-deployment-556dcfc9-v58g  1/1     Running   2 (3m4s ago)  13h
jesus-martinez@jesus-martinez-VirtualBox: ~/docker$ kubectl describe pod jesus
Name:      jesus
Namespace: default
Priority:   0
Service Account: default
Node:      minikube/192.168.49.2
Start Time: Fri, 12 Dec 2025 09:08:13 -0500
Labels:    app=jesus
Annotations: <none>
Status:    Running
IP:        10.244.0.36
IPs:
  IP: 10.244.0.36
Containers:
  jesus:
    Container ID: docker://1caed65b78650b9e376dfb3eaae2710e2d9ce960a7859024813defcf2df25ed
    Image:      jesusmartinezzone/jesus-one:latest
    Image ID:   docker-pullable://jesusmartinezzone/jesus-one@sha256:3723ad30e6dae3963640e9d1b0390851aa6e1a1dce4dd6103889717da95108e4
    Port:      80/TCP
    Host Port:  0/TCP
    Command:    bash
    Args:       -c
               while true; do sleep 3600; done

```

```

Started:      Fri, 12 Dec 2025 13:53:58 -0500
Last State:   Terminated
Reason:       Error
Exit Code:    255
Started:      Fri, 12 Dec 2025 13:48:02 -0500
Finished:     Fri, 12 Dec 2025 13:53:41 -0500
Ready:        True
Restart Count: 2
Environment:  <none>
Mounts:
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-hkrpc (ro)
Conditions:
  Type           Status
  PodReadyToStartContainers  True
  Initialized      True
  Ready            True
  ContainersReady  True
  PodScheduled     True
Volumes:
  kube-api-access-hkrpc:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    Optional:    false
    DownwardAPI: true
  BestEffort:
    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 SandboxChanged 9m4s kubelet Pod sandbox changed, it will be killed and re-created.
  Normal Pulling 9m4s kubelet Pulling image "jesusmartinezzone/jesus-one:latest"
  Normal Pulled 9m4s kubelet Successfully pulled image "jesusmartinezzone/jesus-one:latest" in 870ms (2.25s including waiting). Image size 137997839 bytes.
  Normal Created 9m4s kubelet Created container: jesus

```

3. y por ultimo, creamos una etiqueta para reconocer el pod, creamos un servidor con kubectl expose pod jesus --type=NodePort --port=80 --target-port=80 que permita acceder al pod desde fuera del cluster, y minikube ip nos muestra la ip del nodo de minikube

```

jesus-martinez@jesus-martinez-VirtualBox: ~/docker$ kubectl label pod jesus app=jesus
pod/jesus labeled
jesus-martinez@jesus-martinez-VirtualBox: ~/docker$ kubectl expose pod jesus --type=NodePort --port=80 --target-port=80
service/jesus exposed
jesus-martinez@jesus-martinez-VirtualBox: ~/docker$ kubectl get services
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
jesus     NodePort  10.96.41.146 <none>        80:32728/TCP  8s
kubernetes ClusterIP  10.96.0.1    <none>        443/TCP      6h23m
jesus-martinez@jesus-martinez-VirtualBox: ~/docker$ minikube ip
192.168.49.2
jesus-martinez@jesus-martinez-VirtualBox: ~/docker$

```