PARTE 0: Preparación del entorno local en wsl2

0.1. Instalar k3s

```
root@UbuntuPL:/home/juan# sudo apt update -y && sudo apt upgrade -y
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1,399 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,690 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [227 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [9,716 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2,290 kB]
Get:11 http://es.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [312 kB]
Get:12 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:13 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.8 kB]
Get:14 http://es.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2,418 kB]
47% [14 Packages 22.5 kB/2,418 kB 1%] [10 Packages 1,049 kB/2,290 kB 46%]
```

```
oot@UbuntuPL:/home/juan# sudo apt install -y curl wget git
eading package lists... Done
uilding dependency tree... Done
eading state information... Done
url is already the newest version (8.5.0-2ubuntu10.6).
get is already the newest version (1.21.4-1ubuntu4.1).
get set to manually installed.
he following package was automatically installed and is no longer required:
  libllvm19
se 'sudo apt autoremove' to remove it.
uggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
he following NEW packages will be installed:
  git git-man liberror-perl
 upgraded, 3 newly installed, 0 to remove and 10 not upgraded.
eed to get 4,806 kB of archives.
fter this operation, 24.5 MB of additional disk space will be used.
et:1 http://es.archive.ubuntu.com/ubuntu noble/main amd64 liberror-perl all 0.17029-2 [25.6 kB]
et:2 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 git-man all 1:2.43.0-1ubuntu7.3 [1,100 kB]
et:3 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 git amd64 1:2.43.0-1ubuntu7.3 [3,680 kB]
etched 4,806 kB in 3s (1,856 kB/s)
electing previously unselected package liberror-perl.
(Reading database ... 195115 files and directories currently installed.)
reparing to unpack .../liberror-perl_0.17029-2_all.deb ...
npacking liberror-perl (0.17029-2) ...
electing previously unselected package git-man.
reparing to unpack .../git-man_1%3a2.43.0-1ubuntu7.3_all.deb ...
npacking git-man (1:2.43.0-1ubuntu7.3) ...
electing previously unselected package git.
reparing to unpack .../git_1%3a2.43.0-1ubuntu7.3_amd64.deb ...
npacking git (1:2.43.0-1ubuntu7.3) ...
```

```
root@UbuntuPL:/home/juan# curl -sfL https://get.k3s.io | K3S_KUBECONFIG_MODE="644" sh -
[INFO]  Finding release for channel stable
[INFO]  Using v1.34.3+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.34.3+k3s1/sha256sum-amd64.txt
[INFO]  Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.34.3+k3s1/k3s
[INFO]  Verifying binary download
[INFO]  Installing k3s to /usr/local/bin/k3s
[INFO]  Skipping installation of SELinux RPM
[INFO]  Creating /usr/local/bin/kubectl symlink to k3s
[INFO]  Creating /usr/local/bin/crictl symlink to k3s
[INFO]  Creating /usr/local/bin/ctr symlink to k3s
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s.service
[INFO]  systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.service.
[INFO]  systemd: Starting k3s
root@UbuntuPL:/home/juan#
```

```
root@UbuntuPL:/home/juan# sudo k3s --version
k3s version v1.34.3+k3s1 (48ffa7b6)
go version go1.24.11
root@UbuntuPL:/home/juan#
```

```
root@UbuntuPL:/home/juan# sudo k3s server & sleep 10
[1] 22314
INFO[0000] Starting k3s v1.34.3+k3s1 (48ffa7b6)
INFO[0000] Configuring sqlite3 database connection pooling: maxIdleConns=2, maxOpenConns=0, connMaxLifetime=0s
INFO[0000] Configuring database table schema and indexes, this may take a moment...
INFO[0000] Database tables and indexes are up to date
INFO[0000] Kine available at unix://kine.sock
INFO[0000] Reconciling bootstrap data between datastore and disk
INFO[0000] Password verified locally for node ubuntupl
INFO[0000] certificate CN=ubuntupl signed by CN=k3s-server-ca@1767946964: notBefore=2026-01-09 08:22:44 +0000 UTC notAft
er=2027-01-09 08:26:14 +0000 UTC
INFO[0000] certificate CN=system:kube-proxy signed by CN=k3s-client-ca@1767946964: notBefore=2026-01-09 08:22:44 +0000 U
TC notAfter=2027-01-09 08:26:14 +0000 UTC
INFO[0001] Module overlay was already loaded
INFO[0001] Module nf_conntrack was already loaded
INFO[0001] Module br_netfilter was already loaded
INFO[0001] Module iptable_nat was already loaded
INFO[0001] Module iptable_filter was already loaded
WARN[0001] Failed to load kernel module nft-expr-counter with modprobe
INFO[0001] Creating k3s-cert-monitor event broadcaster
INFO[0001] Running kube-apiserver --advertise-port=6443 --allow-privileged=true --anonymous-auth=false --api-audiences=h
ttps://kubernetes.default.svc.cluster.local,k3s --authorization-mode=Node,RBAC --bind-address=127.0.0.1 --cert-dir=/var/
lib/rancher/k3s/server/tls/temporary-certs --client-ca-file=/var/lib/rancher/k3s/server/tls/client-ca.crt --egress-selec
tor-config-file=/var/lib/rancher/k3s/server/etc/egress-selector-config.yaml --enable-admission-plugins=NodeRestriction -
-enable-aggregator-routing=true --enable-bootstrap-token-auth=true --etcd-servers=unix://kine.sock --kubelet-certificate
-authority=/var/lib/rancher/k3s/server/tls/server-ca.crt --kubelet-client-certificate=/var/lib/rancher/k3s/server/tls/cl
ient-kube-apiserver.crt --kubelet-client-key=/var/lib/rancher/k3s/server/tls/client-kube-apiserver.key --kubelet-preferr
ed-address-types=InternalIP,ExternalIP,Hostname --profiling=false --proxy-client-cert-file=/var/lib/rancher/k3s/server/t
ls/client-auth-proxy.crt --proxy-client-key-file=/var/lib/rancher/k3s/server/tls/client-auth-proxy.key --requestheader-a
llowed-names=system:auth-proxy --requestheader-client-ca-file=/var/lib/rancher/k3s/server/tls/request-header-ca.crt --re
questheader-extra-headers-prefix=X-Remote-Extra- --requestheader-group-headers=X-Remote-Group --requestheader-username-h
eaders=X-Remote-User --secure-port=6444 --service-account-issuer=https://kubernetes.default.svc.cluster.local --service-
account-key-file=/var/lib/rancher/k3s/server/tls/service.key --service-account-signing-key-file=/var/lib/rancher/k3s/ser
```

```
root@UbuntuPL:/home/juan# sudo k3s kubectl get nodes
NAME        STATUS   ROLES           AGE     VERSION
ubuntupl    Ready    control-plane   4m32s   v1.34.3+k3s1
root@UbuntuPL:/home/juan#
```

0.2. Instalar kubectl localmente

```
root@UbuntuPL:/home/juan# curl -LO "https://dl.k8s.io/release/$(curl -L -s
> https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
curl: (2) no URL specified
curl: try 'curl --help' or 'curl --manual' for more information
bash: https://dl.k8s.io/release/stable.txt: No such file or directory
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    972      0 --:--:-- --:--:-- --:--:--   978
100   238  100   238    0     0    700      0 --:--:-- --:--:-- --:--:--   700
root@UbuntuPL:/home/juan#
```

```
root@UbuntuPL:/home/juan# sudo chmod +x kubectl
```

```
root@UbuntuPL:/home/juan# sudo mv kubectl /usr/local/bin/
```

```
root@UbuntuPL:/home/juan# kubectl version --client
Client Version: v1.35.0
Kustomize Version: v5.7.1
root@UbuntuPL:/home/juan#
```

```
root@UbuntuPL:/home/juan# mkdir -p $HOME/.kube
```

```
root@UbuntuPL:/home/juan# sudo cp /etc/rancher/k3s/k3s.yaml $HOME/.kube/config
root@UbuntuPL:/home/juan# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@UbuntuPL:/home/juan# sudo chmod 600 $HOME/.kube/config
```

```
root@UbuntuPL:/home/juan# sudo kubectl get nodes
NAME        STATUS   ROLES           AGE   VERSION
ubuntupl    Ready    control-plane   26m   v1.34.3+k3s1
```

```
root@UbuntuPL:/home/juan# mkdir -p ~/kubernetes-aws-practice
root@UbuntuPL:/home/juan# cd ~/kubernetes-aws-practice
```

PARTE 1: Crear aplicación simple para KUBERNETES

1.1. Carpeta de aplicación

```
root@UbuntuPL:~/kubernetes-aws-practice# mkdir -p ~/kubernetes-aws-practice/app
root@UbuntuPL:~/kubernetes-aws-practice# cd ~/kubernetes-aws-practice/app
```

1.2. Crear aplicación Python con Flask

```
root@UbuntuPL:~/kubernetes-aws-practice/app# cat > app.py << 'EOF'
> #!/usr/bin/env python3
> from flask import Flask, jsonify, send_from_directory
> import os
import socket
from datetime import datetime
import sys
app = Flask(__name__)
# Variables de entorno inyectadas por Kubernetes
POD_NAME = os.getenv('POD_NAME', 'Unknown Pod')
POD_NAMESPACE = os.getenv('POD_NAMESPACE', 'default')
@app.route('/')
def index():
return send_from_directory('.', 'index.html')
@app.route('/pod-info')
def pod_info():
return jsonify({
'pod_name': POD_NAME,
'namespace': POD_NAMESPACE,
'hostname': socket.gethostname(),
'timestamp': datetime.now().isoformat()
})
@app.route('/health')
def health():
return jsonify({'status': 'healthy', 'pod': POD_NAME}), 200
if __name__ == '__main__':
print(f"[{POD_NAME}] Iniciando servidor Flask...", file=sys.stderr)
app.run(host='0.0.0.0', port=5000, debug=False)
EOF
sudo chmod +x app.py
```

1.3. Crear archivo HTML

```
root@UbuntuPL:~/kubernetes-aws-practice/app# cat > index.html << 'EOF'
<!DOCTYPE html>
<html>
<head>
<title>Kubernetes Load Balancer</title>
<style>
body {
font-family: Arial, sans-serif;
display: flex;
justify-content: center;
align-items: center;
min-height: 100vh;
margin: 0;
background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
}
.container {
background: white;
padding: 50px;
border-radius: 10px;
box-shadow: 0 10px 25px rgba(0,0,0,0.2);
text-align: center;
max-width: 500px;
}
h1 {
color: #667eea;
margin: 0 0 30px 0;
}
.info {
background: #f0f0f0;
padding: 20px;
border-radius: 5px;
margin: 20px 0;
EOFtml>t>rror('Error:', e);d-name').textContent = ' Error':name:lizar Ahora</button>
```

1.4. Crear requirements.txt

```
root@UbuntuPL:~/kubernetes-aws-practice/app# cat > requirements.txt << 'EOF'
> Flask==3.0.0
Werkzeug==3.0.0
EOF
```

1.5. Crear Dockerfile ultraligero

```
root@UbuntuPL:~/kubernetes-aws-practice/app# cat > Dockerfile << 'EOF'
> FROM python:3.11-slim
WORKDIR /app
# Copiar archivos
COPY requirements.txt .
COPY app.py .
COPY index.html .
# Instalar dependencias
RUN pip install --no-cache-dir -r requirements.txt
# Ejecutar app
CMD ["python", "app.py"]
EOF
root@UbuntuPL:~/kubernetes-aws-practice/app#
```

PARTE 2:

2.1. Crear Namespace

```
root@UbuntuPL:~/kubernetes-aws-practice/app# cd ~/kubernetes-aws-practice
```

```
root@UbuntuPL:~/kubernetes-aws-practice# cd ~/kubernetes-aws-practice

cat > namespace.yaml << 'EOF'
apiVersion: v1
kind: Namespace
metadata:
  name: load-balancer-demo
  labels:
    name: load-balancer-demo
EOF
root@UbuntuPL:~/kubernetes-aws-practice# kubectl apply -f namespace.yaml
namespace/load-balancer-demo created
```

```
root@UbuntuPL:~/kubernetes-aws-practice# kubectl apply -f namespace.yaml
namespace/load-balancer-demo created
```

```
root@UbuntuPL:~/kubernetes-aws-practice# kubectl get namespaces
NAME                STATUS   AGE
default             Active   52m
kube-node-lease     Active   52m
kube-public         Active   52m
kube-system         Active   52m
load-balancer-demo  Active   2m15s
```

2.2. Crear ConfigMap con archivos de la app

```
root@UbuntuPL:~/kubernetes-aws-practice# cat > configmap.yaml << 'EOF'
apiVersion: v1
kind: ConfigMap
metadata:
name: app-files
namespace: load-balancer-demo
data:
requirements.txt: |
Flask==3.0.0
Werkzeug==3.0.0
app.py: |
#!/usr/bin/env python3
from flask import Flask, jsonify, send_from_directory
import os
import socket
from datetime import datetime
import sys
app = Flask(__name__)
POD_NAME = os.getenv('POD_NAME', 'Unknown Pod')
POD_NAMESPACE = os.getenv('POD_NAMESPACE', 'default')
@app.route('/')
def index():
return send_from_directory('.', 'index.html')
@app.route('/pod-info')
def pod_info():
return jsonify({
'pod_name': POD_NAME,
'namespace': POD_NAMESPACE,
'hostname': socket.gethostname(),
'timestamp': datetime.now().isoformat()
```

```
root@UbuntuPL:~/kubernetes-aws-practice# kubectl apply -f configmap.yaml
configmap/app-files created
root@UbuntuPL:~/kubernetes-aws-practice# kubectl get configmap -n load-balancer-demo
NAME               DATA   AGE
app-files          3      29s
kube-root-ca.crt   1      7m42s
root@UbuntuPL:~/kubernetes-aws-practice#
```

2.3. Crear Deployment con 3 replicas

```
root@UbuntuPL:~/kubernetes-aws-practice# cat > deployment.yaml << 'EOF'
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-app
  namespace: load-balancer-demo
  labels:
    app: web-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web-app
  template:
    metadata:
      labels:
        app: web-app
    spec:
      containers:
        - name: web-app
          image: python:3.11-slim
          command: ["sh", "-c"]
          args:
            - |
              cd /app
              pip install --no-cache-dir -r requirements.txt > /dev/null 2>&1
              python app.py
          ports:
            - containerPort: 5000
              protocol: TCP
          env:
            - name: POD_NAME
EOF          defaultMode: 0755ds: 55ta namespace
```

```
root@UbuntuPL:~/kubernetes-aws-practice# kubectl apply -f deployment.yaml
deployment.apps/web-app created
root@UbuntuPL:~/kubernetes-aws-practice# kubectl get pods -n load-balancer-demo
NAME                       READY   STATUS    RESTARTS   AGE
web-app-65967466dd-drntq   1/1     Running   0          74s
web-app-65967466dd-j7flq   1/1     Running   0          74s
web-app-65967466dd-z6tmf   1/1     Running   0          74s
root@UbuntuPL:~/kubernetes-aws-practice# echo "Esperando a que los pods estén listos..."
Esperando a que los pods estén listos...
root@UbuntuPL:~/kubernetes-aws-practice# kubectl wait --for=condition=ready pod -l app=web-app -n load-balancer-demo --t
imeout=120s
pod/web-app-65967466dd-drntq condition met
pod/web-app-65967466dd-j7flq condition met
pod/web-app-65967466dd-z6tmf condition met
root@UbuntuPL:~/kubernetes-aws-practice# kubectl get pods -n load-balancer-demo
NAME                       READY   STATUS    RESTARTS   AGE
web-app-65967466dd-drntq   1/1     Running   0          2m15s
web-app-65967466dd-j7flq   1/1     Running   0          2m15s
web-app-65967466dd-z6tmf   1/1     Running   0          2m15s
root@UbuntuPL:~/kubernetes-aws-practice#
```

2.4. Crear Service (Load Balancer)

```
root@UbuntuPL:~/kubernetes-aws-practice# cat > service.yaml << 'EOF'
apiVersion: v1
kind: Service
metadata:
  name: web-app-service
  namespace: load-balancer-demo
  labels:
    app: web-app
spec:
  type: LoadBalancer
  selector:
    app: web-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 5000
      name: http
  sessionAffinity: None
EOF
root@UbuntuPL:~/kubernetes-aws-practice# kubectl apply -f service.yaml
service/web-app-service created
root@UbuntuPL:~/kubernetes-aws-practice# kubectl get svc -n load-balancer-demo
NAME              TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
web-app-service   LoadBalancer   10.43.236.142   <pending>     80:30829/TCP   16s
root@UbuntuPL:~/kubernetes-aws-practice# kubectl get svc -n load-balancer-demo web-app-service -o wide
NAME              TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE   SELECTOR
web-app-service   LoadBalancer   10.43.236.142   <pending>     80:30829/TCP   30s   app=web-app
root@UbuntuPL:~/kubernetes-aws-practice#
```

PARTE 3:

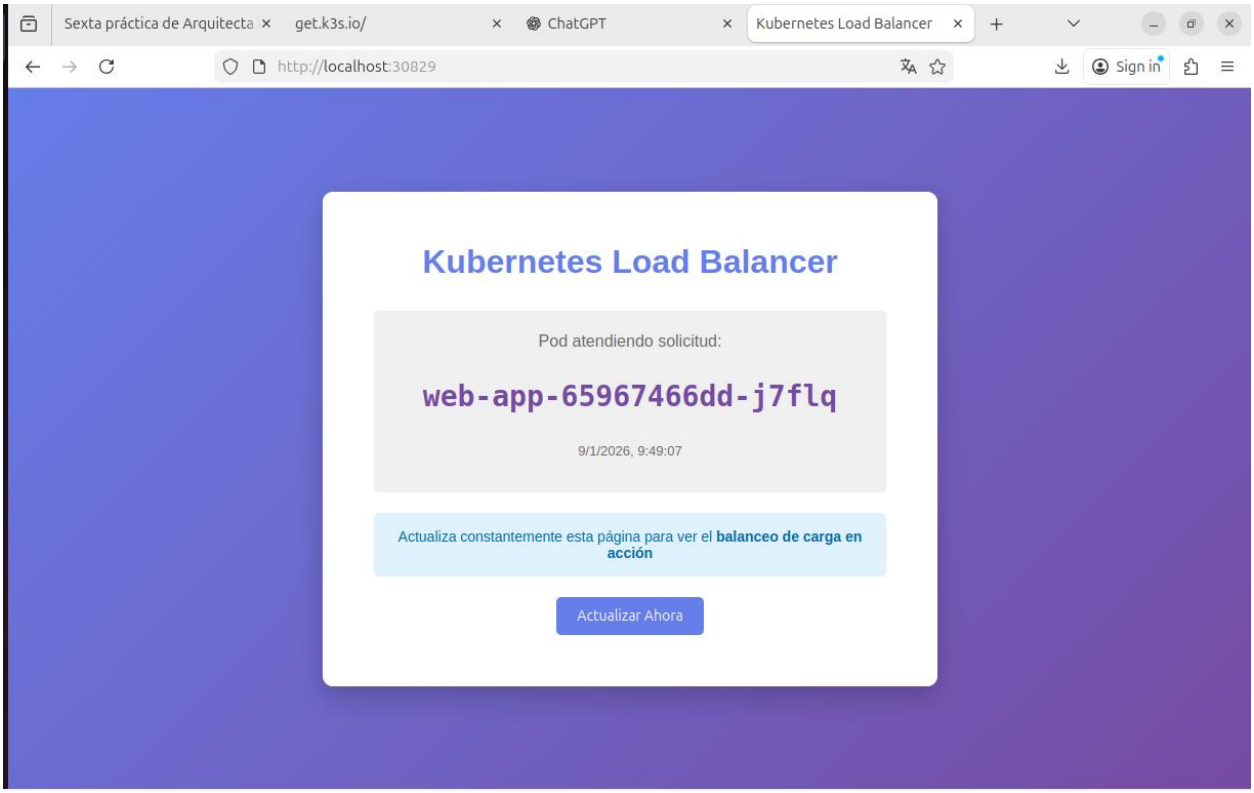
3.1. Levantar la aplicación

```
root@UbuntuPL:~/kubernetes-aws-practice# # Opción 1: Port-forward (recomendado para WSL2)
kubectl port-forward -n load-balancer-demo svc/web-app-service 8080:80
Forwarding from 127.0.0.1:8080 -> 5000
Forwarding from [::1]:8080 -> 5000
```

3.2. Probar balanceo

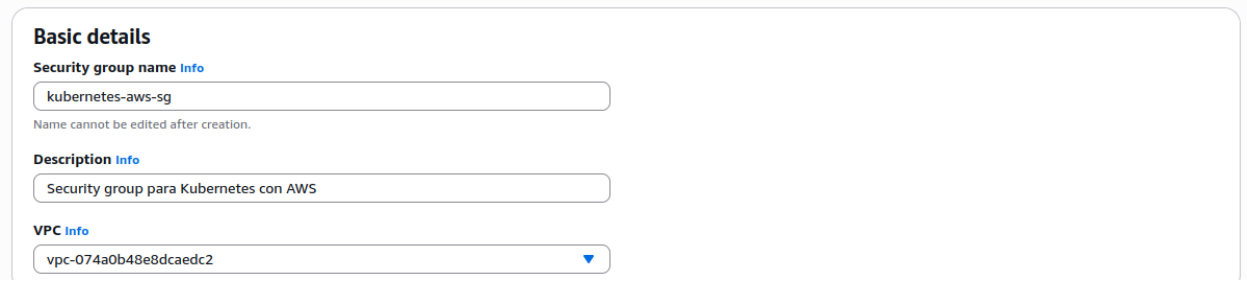
```
root@UbuntuPL:~/kubernetes-aws-practice# # Script para hacer peticiones y ver qué pod responde
for i in {1..10}; do
  echo "Petición $i:"
  curl -s http://localhost:30829/pod-info | python3 -m json.tool | grep pod_name
  sleep 1
done
Petición 1:
    "pod_name": "web-app-65967466dd-j7flq",
Petición 2:
    "pod_name": "web-app-65967466dd-drntq",
Petición 3:
    "pod_name": "web-app-65967466dd-z6tmf",
Petición 4:
    "pod_name": "web-app-65967466dd-z6tmf",
Petición 5:
    "pod_name": "web-app-65967466dd-z6tmf",
Petición 6:
    "pod_name": "web-app-65967466dd-z6tmf",
Petición 7:
    "pod_name": "web-app-65967466dd-drntq",
Petición 8:
    "pod_name": "web-app-65967466dd-z6tmf",
Petición 9:
    "pod_name": "web-app-65967466dd-drntq",
Petición 10:
    "pod_name": "web-app-65967466dd-drntq",
root@UbuntuPL:~/kubernetes-aws-practice#
```

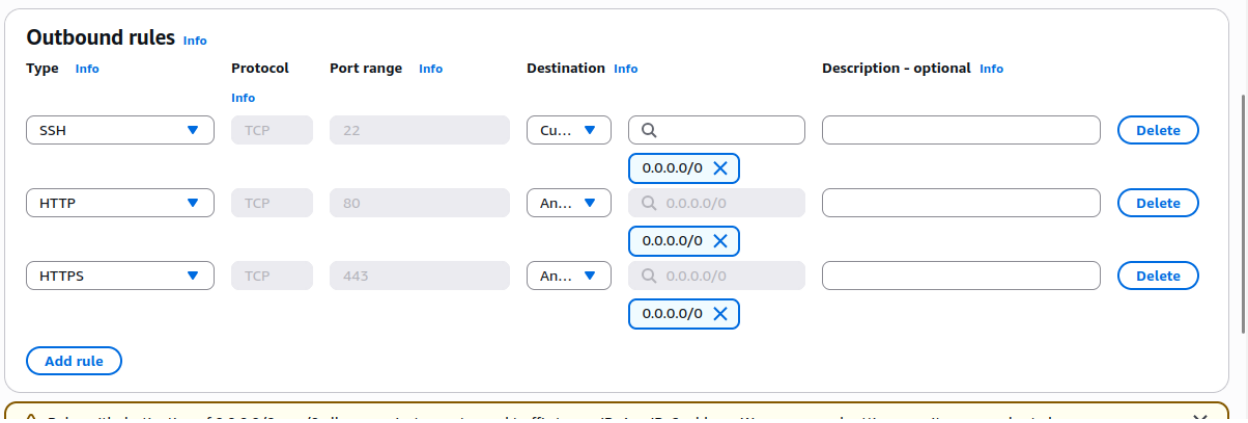
3.3. Verificar en navegador



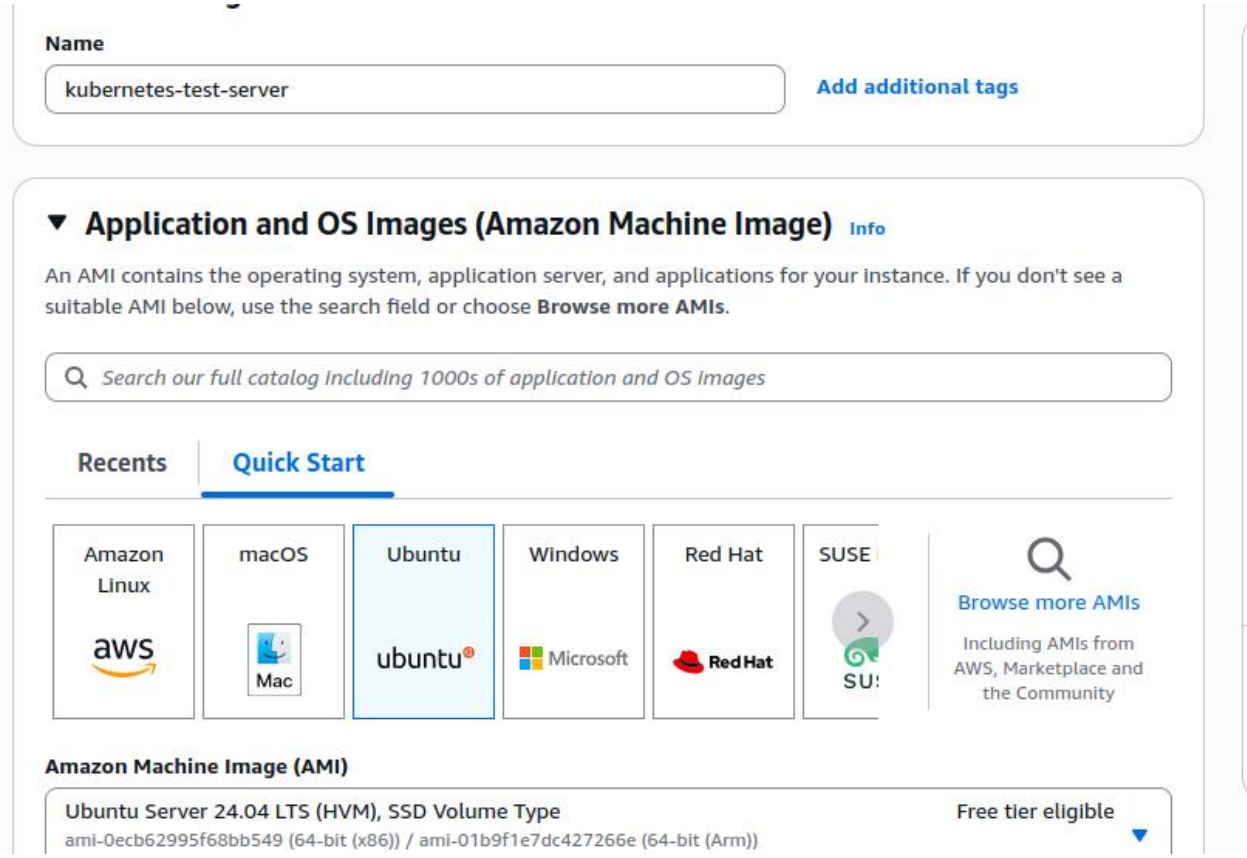
PARTE 4:

4.1. Crear Security Group





4.2. Crear instancia EC2



4.3. Conectar a EC2

```
root@UbuntuPL:~/kubernetes-aws-practice# ssh -i /home/juan/Downloads/clave-kubernetes.pem ubuntu@54.227.10.245
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

 System information as of Fri Jan  9 10:19:54 UTC 2026

  System load:  0.0               Temperature:           -273.1 C
  Usage of /:   25.9% of 6.71GB   Processes:             110
  Memory usage: 23%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 172.31.18.250

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

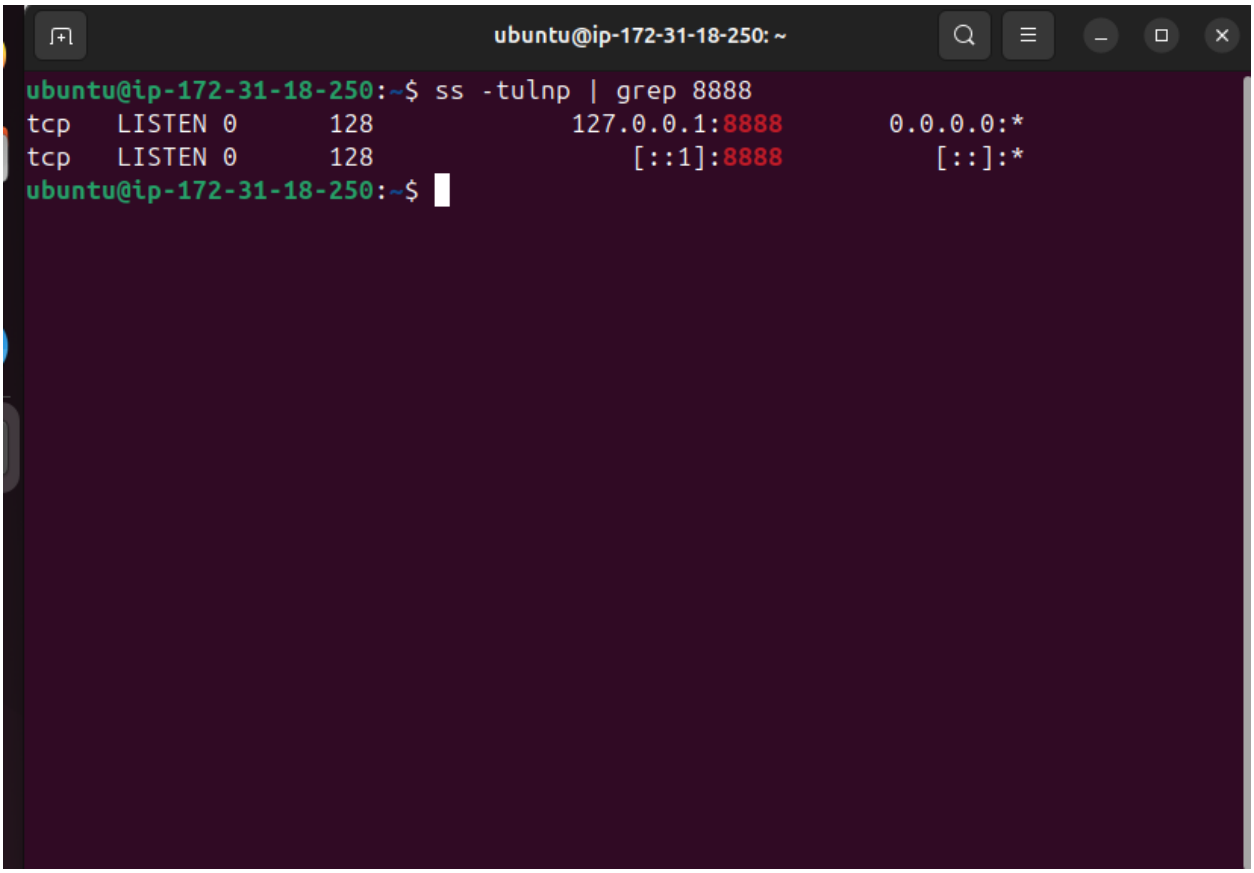
```
ubuntu@ip-172-31-18-250:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1399 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1690 kB]
```

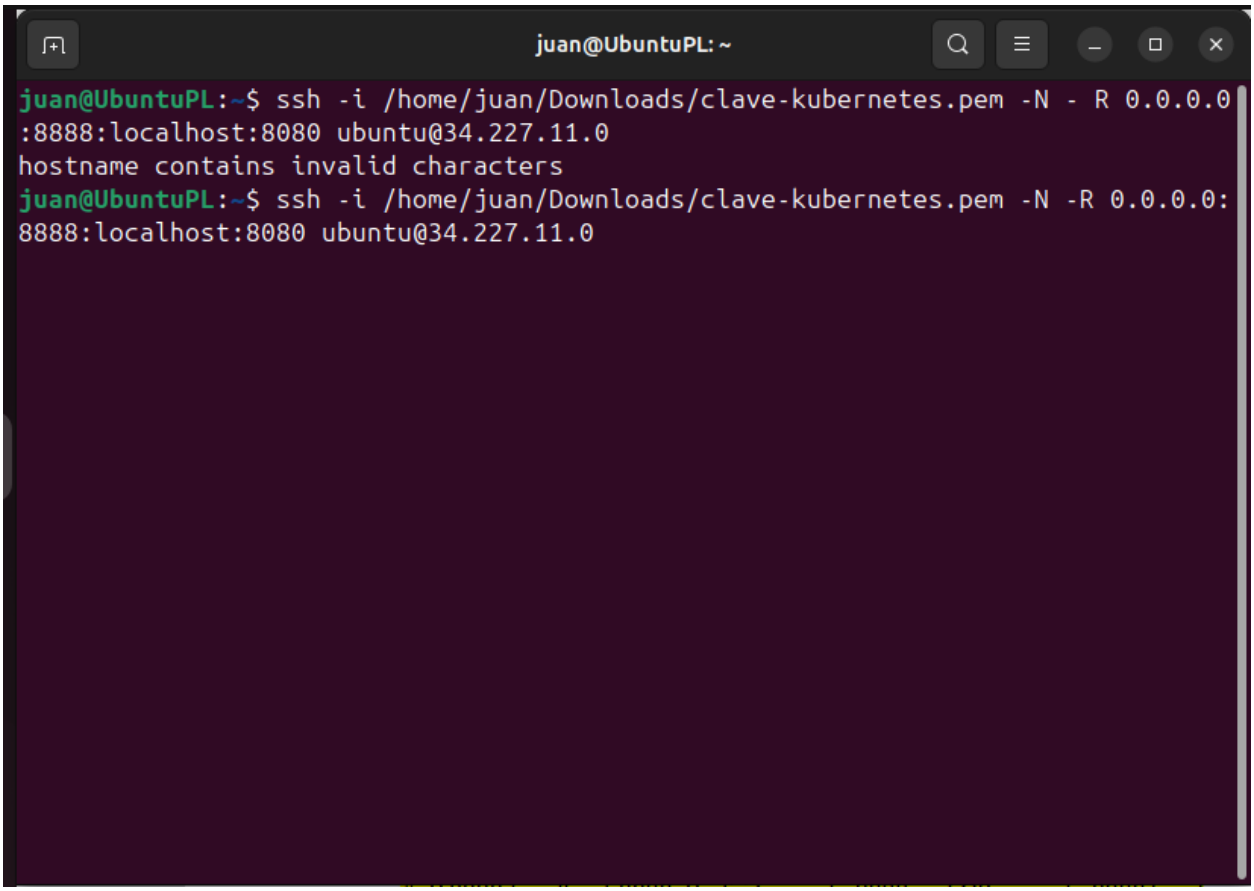
```
ubuntu@ip-172-31-18-250:~$ sudo apt install -y curl wget python3 python3-pip git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
git is already the newest version (1:2.43.0-1ubuntu7.3).
git set to manually installed.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu
  cpp-x86-64-linux-gnu dpkg-dev fakeroot fontconfig-config fonts-dejavu-core fonts-dejavu-mono g++ g++-13
  g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu
  javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom3 libasan8
  libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0
  libdeflate0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libfontconfig1 libgcc-13-dev libgd3
  libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libhwasan0
  libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery libjs-sphinxdoc libjs-underscore liblerc4 liblsan0
  libmpc3 libpython3-dev libpython3-stdlib libpython3.12-dev libpython3.12-minimal libpython3.12-stdlib
  libpython3.12t64 libquadmath0 libsframe1 libsharpyuv0 libstdc++-13-dev libtiff6 libtsan2 libubsan1 libwebp7 libxpm4
  linux-libc-dev linux-tools-common lto-disabled-list make manpages-dev python3-dev python3-minimal python3-wheel
  python3.12 python3.12-dev python3.12-minimal rpcsvc-proto zlib1g-dev
Suggested packages:
  binutils-doc gprofng-gui bzip2-doc cpp-doc gcc-13-locales cpp-13-doc debian-keyring g++-multilib g++-13-multilib
  gcc-13-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc gcc-13-multilib gdb-x86-64-linux-gnu
  apache2 | lighttpd | httpd glibc-doc bzr libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec
  libheif-plugin-jpegdec libheif-plugin-jpegenc libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e
  libheif-plugin-svtenc libstdc++-13-doc make-doc python3-doc python3-tk python3-venv python3.12-venv python3.12-doc
```

```
ubuntu@ip-172-31-18-250:~$ mkdir -p ~/kubernetes-test
ubuntu@ip-172-31-18-250:~$
```

PARTE 5:

5.1. Crear tuner SSH que expone el LoadBalancer


ubuntu@ip-172-31-18-250: ~
ubuntu@ip-172-31-18-250:~$ ss -tulnp | grep 8888
tcp LISTEN 0 128 127.0.0.1:8888 0.0.0.0:*
tcp LISTEN 0 128 [::1]:8888 [::]:*
ubuntu@ip-172-31-18-250:~$



5.2. Crear script de prueba

```
ubuntu@ip-172-31-18-250:~$ cat > ~/test-kubernetes-lb.sh << 'EOF'
> #!/bin/bash
echo "=== Prueba Kubernetes Load Balancer desde AWS ==="
echo ""
declare -A pods_count
for i in {1..15}; do
response=$(curl -s http://localhost:8888/pod-info)
pod_name=$(echo $response | python3 -c "import sys, json;
print(json.load(sys.stdin)['pod_name'])" 2>/dev/null)
if [ -z "$pod_name" ]; then
pod_name="ERROR"
fi
echo "Petición $i → Pod: $pod_name"
pods_count[$pod_name]=$((${pods_count[$pod_name]:-0} + 1))
sleep 1
done
echo ""
echo "=== Resumen Balanceo ==="
for pod in "${!pods_count[@]}"; do
echo "Pod $pod: ${pods_count[$pod]} peticiones"
done
EOF
ubuntu@ip-172-31-18-250:~$ sudo chmod +x ~/test-kubernetes-lb.sh
```

```
ubuntu@ip-172-31-18-250:~$ ~/test-kubernetes-lb.sh
=== Prueba Kubernetes Load Balancer desde AWS ===

Petición 1 → Pod: ERROR
Petición 2 → Pod: ERROR
Petición 3 → Pod: ERROR
Petición 4 → Pod: ERROR
Petición 5 → Pod: ERROR
Petición 6 → Pod: ERROR
Petición 7 → Pod: ERROR
Petición 8 → Pod: ERROR
Petición 9 → Pod: ERROR
Petición 10 → Pod: ERROR
Petición 11 → Pod: ERROR
```

```
=== Resumen Balanceo ===
Pod ERROR: 15 peticiones
```

PARTE 6:

6.1. Escalar a 5 replicas

```
juan@UbuntuPL:~$ sudo kubectl scale deployment web-app -n load-balancer-demo --replicas=5

[sudo] password for juan:
deployment.apps/web-app scaled
```

```
juan@UbuntuPL:~$ sudo kubectl get pods -n load-balancer-demo
NAME                          READY   STATUS    RESTARTS        AGE
web-app-65967466dd-99bsn      1/1     Running   0               63s
web-app-65967466dd-d9xrh      1/1     Running   0               63s
web-app-65967466dd-drntq      1/1     Running   6 (3d19h ago)   14d
web-app-65967466dd-j7flq      1/1     Running   6 (3d19h ago)   14d
web-app-65967466dd-z6tmf      1/1     Running   6 (3d19h ago)   14d
```

```
juan@UbuntuPL:~$ sudo kubectl wait --for=condition=ready pod -l app=web-app -n load-balancer-demo --timeout=120s
pod/web-app-65967466dd-99bsn condition met
pod/web-app-65967466dd-d9xrh condition met
pod/web-app-65967466dd-drntq condition met
pod/web-app-65967466dd-j7flq condition met
pod/web-app-65967466dd-z6tmf condition met
```

6.2. Probar balanceo (La máquina no funciona bien, el puerto 8888 lo está escuchando, el túnel está conectado, pero no acciona a nada)

```
ubuntu@ip-172-31-18-250:~$ ~/test-kubernetes-lb.sh
=== Prueba Kubernetes Load Balancer desde AWS ===
```

PARTE 7:

7.1. Estado pods

TODOS

```
juan@UbuntuPL:~$ sudo kubectl get pods -n load-balancer-demo
[sudo] password for juan:
NAME                          READY   STATUS    RESTARTS        AGE
web-app-65967466dd-99bsn      1/1     Running   0               44m
web-app-65967466dd-d9xrh      1/1     Running   0               44m
web-app-65967466dd-drntq      1/1     Running   6 (3d20h ago)   14d
web-app-65967466dd-j7flq      1/1     Running   6 (3d20h ago)   14d
web-app-65967466dd-z6tmf      1/1     Running   6 (3d20h ago)   14d
```

11111

```
juan@UbuntuPL:~$ sudo kubectl logs -n load-balancer-demo web-app-65967466dd-99bs
n
[web-app-65967466dd-99bsn] Iniciando servidor Flask...
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.42.0.60:5000
Press CTRL+C to quit
10.42.0.1 - - [23/Jan/2026 09:30:09] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 09:30:14] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 09:30:19] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 09:30:22] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 09:30:24] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 09:30:29] "GET /health HTTP/1.1" 200 -
```

TIEMPO REAL

```
juan@UbuntuPL:~$ sudo kubectl logs -n load-balancer-demo -l app=web-app -f
10.42.0.1 - - [23/Jan/2026 10:16:02] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:03] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:08] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:12] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:13] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:18] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:22] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:23] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:28] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:32] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:02] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:04] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:09] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:12] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:14] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:19] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:22] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:24] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:29] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:32] "GET /health HTTP/1.1" 200 -
10.42.0.1 - - [23/Jan/2026 10:16:00] "GET /health HTTP/1.1" 200 -
```

7.2. Ver estadisticas

-CPU MEMORIA

```
juan@UbuntuPL:~$ sudo kubectl top pods -n load-balancer-demo
NAME                          CPU(cores)   MEMORY(bytes)
web-app-65967466dd-99bsn      2m           22Mi
web-app-65967466dd-d9xrh      1m           22Mi
web-app-65967466dd-drntq      1m           23Mi
web-app-65967466dd-j7flq      2m           22Mi
web-app-65967466dd-z6tmf      1m           23Mi
juan@UbuntuPL:~$
```

-NODOS

```
juan@UbuntuPL:~$ sudo kubectl top nodes
NAME       CPU(cores)   CPU(%)   MEMORY(bytes)   MEMORY(%)
ubuntupl   359m         8%       3191Mi          81%
juan@UbuntuPL:~$
```

-EVENTO

```
juan@UbuntuPL:~$ sudo kubectl get events -n load-balancer-demo
LAST SEEN   TYPE     REASON              OBJECT                              MESSAGE
49m         Normal   Scheduled           pod/web-app-65967466dd-99bsn        Success
fully assigned load-balancer-demo/web-app-65967466dd-99bsn to ubuntupl
49m         Normal   Pulled              pod/web-app-65967466dd-99bsn        Contain
er image "python:3.11-slim" already present on machine
49m         Normal   Created             pod/web-app-65967466dd-99bsn        Created
 container: web-app
49m         Normal   Started             pod/web-app-65967466dd-99bsn        Started
 container web-app
49m         Normal   Scheduled           pod/web-app-65967466dd-d9xrh        Success
fully assigned load-balancer-demo/web-app-65967466dd-d9xrh to ubuntupl
49m         Normal   Pulled              pod/web-app-65967466dd-d9xrh        Contain
er image "python:3.11-slim" already present on machine
49m         Normal   Created             pod/web-app-65967466dd-d9xrh        Created
 container: web-app
49m         Normal   Started             pod/web-app-65967466dd-d9xrh        Started
 container web-app
49m         Normal   SuccessfulCreate    replicaset/web-app-65967466dd       Created
 pod: web-app-65967466dd-d9xrh
49m         Normal   SuccessfulCreate    replicaset/web-app-65967466dd       Created
 pod: web-app-65967466dd-99bsn
```

7.3. Ver detalles del employment

-Informacion completa

```
juan@UbuntuPL:~$ sudo kubectl describe deployment web-app -n load-balancer-demo
Name:                   web-app
Namespace:              load-balancer-demo
CreationTimestamp:      Fri, 09 Jan 2026 09:23:33 +0000
Labels:                 app=web-app
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=web-app
Replicas:               5 desired | 5 updated | 5 total | 5 available | 0 unavai
lable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=web-app
  Containers:
   web-app:
    Image:      python:3.11-slim
    Port:       5000/TCP
    Host Port:  0/TCP
    Command:
      sh
      -c
    Args:
```

-Historial

```
juan@UbuntuPL:~$ sudo kubectl rollout history deployment web-app -n load-balance
r-demo
deployment.apps/web-app
REVISION  CHANGE-CAUSE
1         <none>

juan@UbuntuPL:~$
```

PARTE 8:

8.1. Liampiar kubernetes

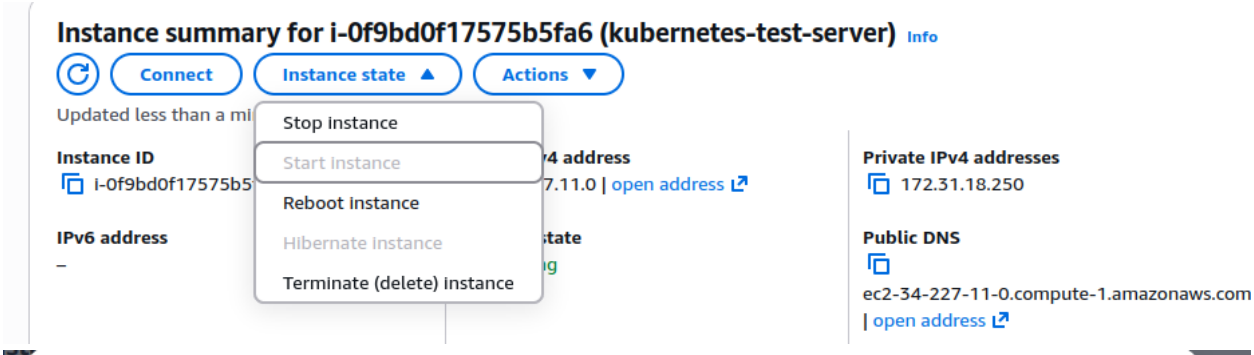
-Eliminar NAMESPACE

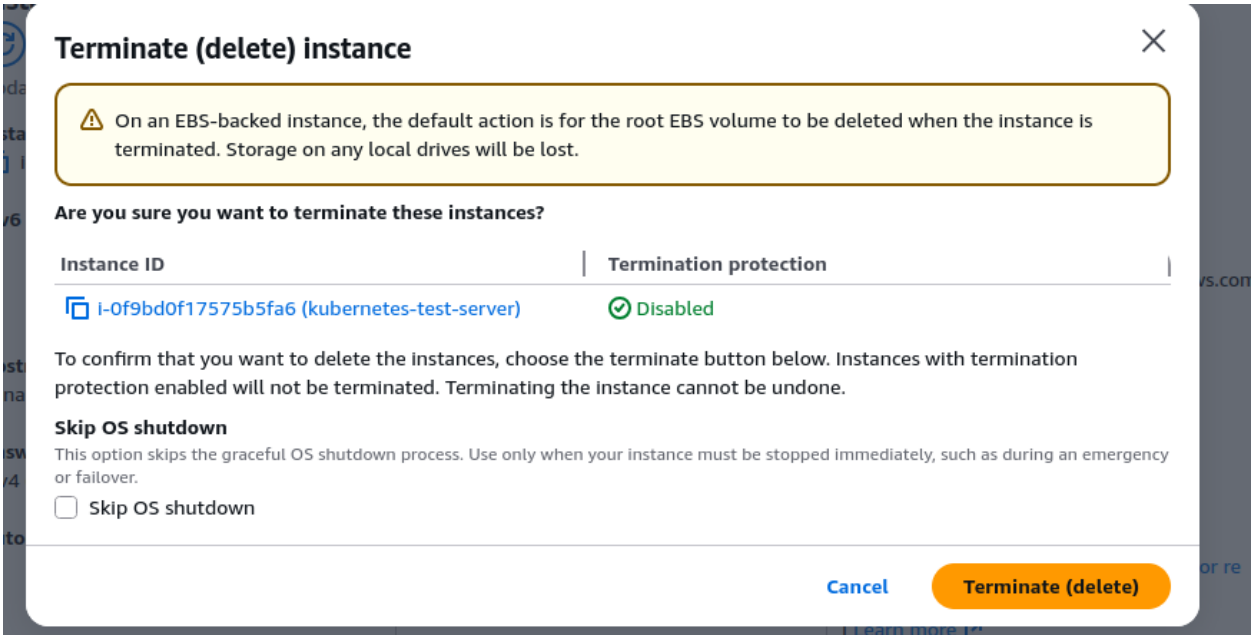
```
juan@UbuntuPL:~$ sudo kubectl delete namespace load-balancer-demo
namespace "load-balancer-demo" deleted
```

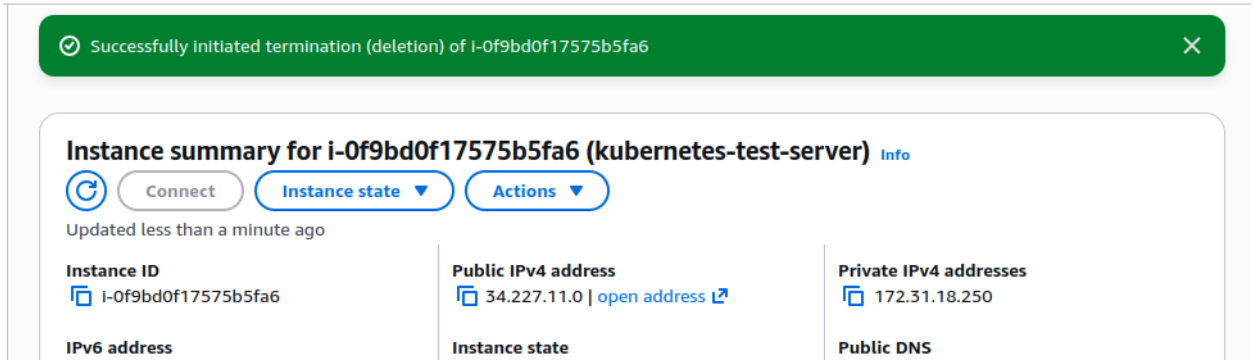
-Verificar

```
juan@UbuntuPL:~$ sudo kubectl get namespaces
NAME                 STATUS        AGE
default              Active        14d
kube-node-lease      Active        14d
kube-public          Active        14d
kube-system          Active        14d
load-balancer-demo   Terminating   14d
```

8.2. Eliminar Instacia EC2







8.3. Detener K3S

-PAUSAR Y ELIMINAR

```
juan@UbuntuPL:~$ sudo systemctl stop k3s
juan@UbuntuPL:~$ sudo /usr/local/bin/k3s-uninstall.sh
+ id -u
+ [ 0 -eq 0 ]
+ K3S_DATA_DIR=/var/lib/rancher/k3s
+ /usr/local/bin/k3s-killall.sh
+ [ -s /etc/systemd/system/k3s.service ]
+ basename /etc/systemd/system/k3s.service
+ systemctl stop k3s.service
+ [ -x /etc/init.d/k3s* ]
+ killtree
+ kill -9
+ do_unmount_and_remove /run/k3s
+ set +x
+ do_unmount_and_remove /var/lib/kubelet/pods
+ set +x
+ do_unmount_and_remove /var/lib/kubelet/plugins
+ set +x
+ do_unmount_and_remove /run/netns/cni-
+ set +x
+ + ip netns show
+ xargs -r -t -n 1 ip netns delete
grep cni-
+ remove_interfaces
+ ip link show
```