# Manual de Implementación: Sistema de Escalado Automático con KEDA, Kafka (Strimzi) y Eventos Wikimedia

## Deben revisar los archivos adjuntos en el foro

**Dockerfile**

## Implementación Paso a Paso

**Configuración del Entorno Local**

**Instalación de Herramientas en Windows**

1. Instalar Chocolatey (si no está instalado)

Set-ExecutionPolicy Bypass -Scope Process -Force;

iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))

2. Instalar Docker Desktop

choco install docker-desktop -y

3. Instalar kubectl

choco install kubernetes-cli -y

4. Instalar minikube

choco install minikube -y

5. Instalar Helm

choco install kubernetes-helm -y

**Iniciar Minikube**

Asegurar que Docker Desktop está corriendo

Iniciar minikube con suficientes recursos

minikube start --driver=docker --cpus=4 --memory=8g --disk-size=50g

Verificar que minikube está funcionando

kubectl get nodes

minikube status

## Construcción de Imágenes Docker

**Preparar la Estructura de Archivos**

Crear estructura de directorios

mkdir -p wikimedia-demo/producer

mkdir -p wikimedia-demo/consumer

Cambiar al directorio principal

cd wikimedia-demo

Crea los archivos según las secciones anteriores en sus respectivas ubicaciones.

**Construir y Publicar Imágenes**

Construir imágenes

cd producer

docker build -t diexlo/wikimedia-producer:1.0 .

cd ../consumer

docker build -t diexlo/wikimedia-consumer:1.0 .

cd ..

Publicar imágenes

docker push diexlo/wikimedia-producer:1.0

docker push diexlo/wikimedia-consumer:1.0

## Implementación de Strimzi (Kafka)

**Eliminar todo**

**kubectl delete namespace $(kubectl get ns --no-headers | grep -vE 'kube-system|kube-public|kube-node-lease|default' | awk '{print $1}')**

**kubectl delete all --all --all-namespaces**

**kubectl delete namespace $(kubectl get ns -o name | grep -vE 'kube-system|kube-public|kube-node-lease|default')**

**kubectl delete pvc --all --all-namespaces**

**kubectl delete pv --all**

**kubectl delete secrets --all --all-namespaces**

**kubectl delete configmap --all --all-namespaces**

**kubectl delete ingress --all --all-namespaces**

**kubectl delete crd $(kubectl get crd | grep strimzi | awk '{print $1}')**

**kubectl delete namespace $(kubectl get ns --no-headers | awk '{print $1}')**

**minikube delete --purge**

**Crear Namespace para Kafka**

kubectl create namespace kafka

**Instalar Strimzi Operator**

Añadir repositorio Helm de Strimzi

helm repo add strimzi https://strimzi.io/charts/

helm repo update

Instalar Strimzi Operator

helm install strimzi-kafka-operator strimzi/strimzi-kafka-operator --namespace kafka --create-namespace

ver logs

kubectl logs -n kafka strimzi-cluster-operator-76b947897f-64gnb –follow

corrige

**helm install strimzi strimzi/strimzi-kafka-operator --version** 0.45.0 **-n kafka --create-namespace**

**desintalar**

**helm uninstall strimzi -n kafka**

Verificar que el operador esté funcionando

kubectl get pods -n kafka -w

**Desplegar un Clúster Kafka**

Crea un archivo llamado kafka-cluster.yaml:

Aplica la configuración:

kubectl apply -f kafka-cluster.yaml -n kafka

kubectl delete -f kafka-cluster.yaml -n kafka

Ver el estado de Kafka

kubectl get kafka -A

Monitorea la creación de los pods de Kafka

kubectl get pods -n kafka -w

**Crear Topic de Kafka**

Crea un archivo llamado kafka-topic.yaml:

Aplica la configuración:

kubectl apply -f kafka-topic.yaml -n kafka

kubectl delete -f kafka-topic.yaml -n kafka

Verifica que el topic se haya creado

kubectl get kafkatopics -n kafka

ver detalle topic

kubectl describe kafka my-cluster -n kafka

## Implementación de KEDA

**Crear Namespace para KEDA**

kubectl create namespace keda

**Instalar KEDA usando Helm**

Agregar el repositorio Helm de KEDA

helm repo add kedacore https://kedacore.github.io/charts

helm repo update

Instalar KEDA

helm install keda kedacore/keda --namespace keda

Verificar que KEDA esté instalado correctamente

kubectl get pods -n keda

**Despliegue de Aplicaciones**

**Crear Namespace para la Aplicación**

kubectl Wikime namespace Wikimedia-demo

**Desplegar el Productor**

Crea un archivo llamado producer-deployment.yaml:

Aplica la configuración:

kubectl apply -f producer-deployment.yaml

**Desplegar el Consumidor**

Crea un archivo llamado consumer-deployment.yaml:

Aplica la configuración:

kubectl apply -f consumer-deployment.yaml

**Configuración del Escalado Automático**

Crea un archivo llamado keda-scaledobject.yaml:

Aplica la configuración:

kubectl apply -f keda-scaledobject.yaml

**Verificación y Pruebas**

**Verificar Pods en Funcionamiento**

En una terminal, monitorea los pods

kubectl get pods -n wikimedia-demo -w

**Verificar los Logs del Productor**

kubectl logs -f deployment/wikimedia-producer -n wikimedia-demo

**Verificar los Logs de los Consumidores (Solo sobre powershell)**

Sustituye POD\_NAME con el nombre real de uno de los pods consumidores

POD\_NAME=$(kubectl get pods -n wikimedia-demo -l app=wikimedia-consumer -o jsonpath='{.items[0].metadata.name}')

kubectl logs -f $POD\_NAME -n wikimedia-demo

**Verificar el Escalado Automático**

Verificar HPA creado por KEDA

kubectl get hpa -n wikimedia-demo

Ver detalles del ScaledObject

kubectl describe scaledobject wikimedia-consumer-scaler -n wikimedia-demo

**Pruebas de Carga y Escalado**

Escalar el productor para generar más mensajes

kubectl scale deployment/wikimedia-producer --replicas=2 -n wikimedia-demo

Monitorear el escalado de los consumidores

kubectl get pods -n wikimedia-demo -w

**Prueba de Descalado a Cero**

Pausar el productor

kubectl scale deployment/wikimedia-producer --replicas=0 -n wikimedia-demo

Monitorear el descalado de los consumidores

kubectl get pods -n wikimedia-demo -w

## Resolución de Problemas

**Problemas con la Conexión a Kafka**

Verificar que los servicios de Kafka estén disponibles

kubectl get services -n kafka

Verificar que el topic exista

kubectl get kafkatopics -n kafka

Probar la conectividad entre namespaces

kubectl run test-kafka -n wikimedia-demo --rm -i --tty --image=quay.io/strimzi/kafka:0.33.0-kafka-3.5.0 -- bin/kafka-topics.sh --bootstrap-server my-cluster-kafka-bootstrap.kafka.svc.cluster.local:9092 --list

**Problemas con KEDA**

Verificar los logs del operador KEDA

kubectl logs -l app=keda-operator -n keda

Verificar el estado del ScaledObject

kubectl describe scaledobject wikimedia-consumer-scaler -n wikimedia-demo

Verificar que el HPA se haya creado correctamente

kubectl get hpa -n wikimedia-demo

kubectl describe hpa -n wikimedia-demo

**Problemas con los Pods Consumidores**

Verificar logs

kubectl logs -l app=wikimedia-consumer -n wikimedia-demo

Verificar configuración

kubectl describe deployment wikimedia-consumer -n wikimedia-demo

**Falta de Recursos en Minikube**

Detener minikube

minikube stop

Reiniciar con más recursos

minikube start --driver=docker --cpus=6 --memory=10g --disk-size=30g

**Limpieza del Entorno**

Eliminar los recursos de la aplicación

kubectl delete namespace wikimedia-demo

Eliminar KEDA

kubectl delete namespace keda

Eliminar Kafka

kubectl delete namespace kafka

Detener minikube

minikube stop

**Referencias**

* [Documentación oficial de KEDA](https://keda.sh/docs/)
* [Documentación de Strimzi](https://strimzi.io/documentation/)
* [Kafka JS Client](https://kafka.js.org/)
* [Stream de eventos de Wikimedia](https://wikitech.wikimedia.org/wiki/Event_Platform/EventStreams)
* [Documentación de Kubernetes](https://kubernetes.io/docs/)
* [Minikube Documentation](https://minikube.sigs.k8s.io/docs/)