

Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Ingeniería en Ciencias y Sistemas  
Sistemas Operativos 1  
Junio 2024



# Manual de Usuario

Grupo 12

# Manual de Usuario

## Objetivo general

El objetivo general de este manual de usuario es proporcionar una guía detallada y accesible para los usuarios del sistema "2024 Weather Tweets", permitiéndoles entender cómo interactuar efectivamente con las herramientas clave del proyecto y aprovechar al máximo su funcionalidad.

## Objetivos específicos

Descripción y Acceso a Locust:

1. Proporcionar instrucciones claras sobre cómo acceder a la herramienta Locust para simular tráfico concurrente y enviar datos al sistema distribuido.
2. Detallar los parámetros de configuración necesarios en Locust para realizar pruebas de carga efectivas.

Descripción del Dashboard en Grafana:

1. Explicar cómo acceder al dashboard de Grafana para visualizar y monitorear los mensajes y métricas del sistema distribuido.
2. Mostrar datos relevantes almacenados en Redis.

Descripción de Estadísticas de Kepler:

1. Informar sobre cómo acceder a las estadísticas proporcionadas por Kepler para monitorear el consumo de energía y las emisiones de carbono por pod en Kubernetes.
2. Explicar cómo interpretar y utilizar las métricas de Kepler para optimizar el uso de recursos y mejorar la sostenibilidad ambiental del sistema.

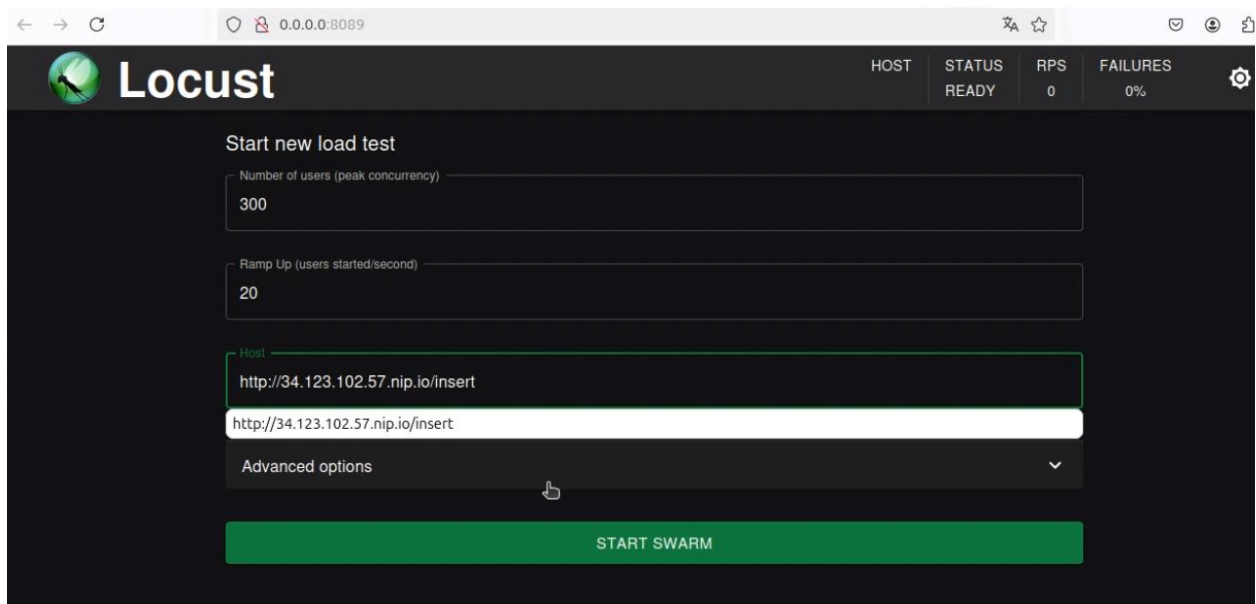
## Locust

Locust es utilizado para simular cargas concurrentes en el sistema. En el proyecto se implementó para enviar tráfico concurrente con lo cual se insertarán los registros (tweets).

Acceso:

Levantar Locust con la línea de comandos en el directorio en el que se encuentran los archivos data.json y trafic.py :

*locust -f trafic.py*



The screenshot shows the Locust web interface in a browser window. The address bar shows '0.0.0.0:8089'. The interface has a dark theme. At the top, there's a header with the Locust logo and a table with columns: HOST, STATUS, RPS, and FAILURES. The STATUS is 'READY', RPS is '0', and FAILURES is '0%'. Below the header, there's a section titled 'Start new load test'. It contains three input fields: 'Number of users (peak concurrency)' with the value '300', 'Ramp Up (users started/second)' with the value '20', and 'Host' with the value 'http://34.123.102.57.nip.io/insert'. There's also an 'Advanced options' dropdown menu. At the bottom, there's a large green button labeled 'START SWARM'.

En los campos respectivos colocar:

- Número de usuarios
- Cantidad de peticiones por segundo
- Host (para este proyecto ingresaremos la URL generada en las reglas de entrada de Ingress 34.123.102.57.nip.io/insert )

## Reportería

El dashboard de Grafana consiste de dos paneles:

- *Porcentaje de tweets según país*
- *Cantidad de mensajes según país*

El dashboard en Grafana se diseñó para mostrar datos almacenados en Redis, utilizando tablas hash para almacenar un contador por país y un contador total de mensajes.

Acceso público: <http://34.132.170.139:3000/>



## Kepler

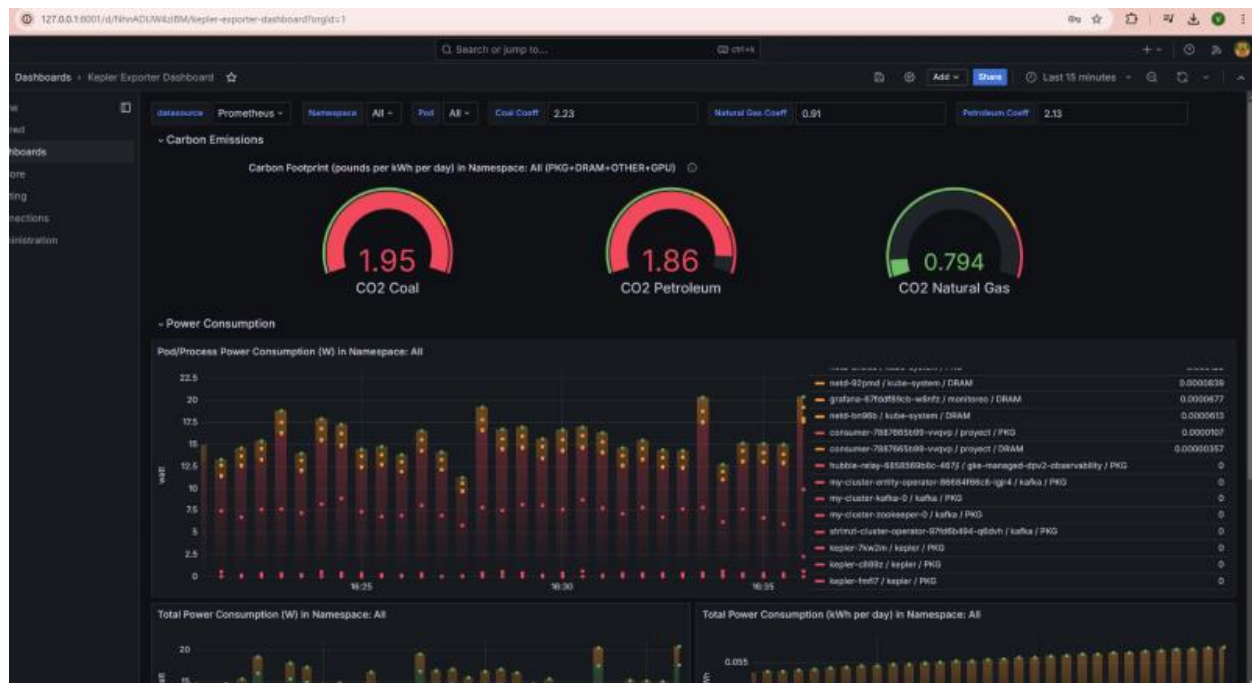
Para la visualización de estadísticas y monitoreo de sustentabilidad, se implementó un dashboard de Grafana con las estadísticas obtenidas por el servicio de Kepler.

- Visualización del consumo de energía por pod y por espacio de nombre
- Visualización de emisiones de carbono por pod y por espacio de nombre

Acceso:

Se levanta la aplicación de Grafana localmente con:

```
kubectll port-forward svc/prometheus-grafana 8001:80 -n monitoring
```



## Glosario de términos

### A

**API (Interfaz de Programación de Aplicaciones):** Conjunto de definiciones y protocolos que permite la comunicación entre diferentes componentes de software.

### C

**Cluster de Kubernetes:** Conjunto de máquinas (nodos) que ejecutan aplicaciones en contenedores y son gestionadas por Kubernetes.

**Concurrente:** Situación en la cual múltiples tareas o procesos se ejecutan al mismo tiempo.

**Contenedor:** Unidad estándar de software que empaqueta el código y todas sus dependencias para que la aplicación se ejecute de manera rápida y confiable en diferentes entornos informáticos.

### D

**Dashboard:** Panel de control interactivo que visualiza datos y métricas clave de un sistema o aplicación.

Distribuido: Sistema en el cual los componentes se encuentran en diferentes ubicaciones, pero funcionan como un solo sistema unificado.

## G

Git: Sistema de control de versiones que permite a múltiples personas trabajar en el mismo código sin sobrescribir los cambios de los demás.

GitHub: Plataforma de alojamiento para control de versiones y colaboración de proyectos que usa Git.

gRPC: Framework de comunicación remota que utiliza HTTP/2 para el transporte, Protobuf para la serialización de datos y proporciona alta eficiencia en las comunicaciones entre servicios.

## H

Helm: Herramienta para gestionar aplicaciones en Kubernetes mediante la utilización de "charts", que son paquetes preconfigurados de recursos de Kubernetes.

HPC (Computación de Alto Rendimiento): Uso de supercomputadoras y clústeres de computadoras para realizar cálculos complejos a gran velocidad.

## I

Ingress Controller: Componente de Kubernetes que gestiona el acceso externo a los servicios en un clúster, generalmente a través de HTTP/HTTPS.

## J

JSON (JavaScript Object Notation): Formato de texto ligero para el intercambio de datos que es fácil de leer y escribir para los humanos y fácil de parsear y generar para las máquinas.

## K

Kafka: Plataforma distribuida de transmisión de datos en tiempo real, diseñada para manejar flujos de datos de gran volumen con baja latencia.

Kepler: Herramienta utilizada para visualizar el consumo de energía y las emisiones de carbono en un clúster de Kubernetes.

Kubernetes: Plataforma de código abierto para automatizar el despliegue, escalado y operación de aplicaciones en contenedores.

## L

Load Balancer (Balanceador de Carga): Dispositivo o software que distribuye el tráfico de red o las cargas de trabajo entre varios servidores para asegurar una carga equilibrada y evitar la sobrecarga de un solo recurso.

Locust: Herramienta de código abierto para realizar pruebas de carga distribuidas y evaluar el rendimiento de aplicaciones web.

## M

MongoDB: Base de datos NoSQL orientada a documentos que almacena datos en formato JSON.

Microservicios: Estilo arquitectónico que estructura una aplicación como un conjunto de servicios pequeños y autónomos, cada uno ejecutando un único proceso y comunicándose a través de interfaces bien definidas.

## N

Namespace: Método para organizar y separar objetos dentro de un clúster de Kubernetes, permitiendo una gestión más eficiente y segura de los recursos.

## P

Pod: Unidad básica de ejecución en Kubernetes que encapsula uno o más contenedores, compartiendo la misma red y almacenamiento.

## R

Redis: Base de datos en memoria de estructura de datos clave-valor, utilizada para almacenamiento en caché, cola de mensajes y otras tareas de almacenamiento rápido.

RPC (Remote Procedure Call): Protocolo que un programa puede usar para solicitar servicios de un programa ubicado en otra computadora en una red.

## S

Strimzi: Proyecto de código abierto que facilita la ejecución de Apache Kafka en Kubernetes mediante la utilización de operadores.

W

WebSocket: Protocolo de comunicación que proporciona un canal full-duplex sobre una única conexión TCP, permitiendo la comunicación interactiva entre el navegador y el servidor.