

Base de Datos II (IC4302) – Semestre 1, 2023

Tarea Corta #1 – Observability

Jennifer Alvarado Brenes – 2020124171

Luis Diego Delgado Muñoz – 2020030408

Esteven Fernandez Hernandez – 2016072253

Erick Madrigal Zavala – 2018146983

David Suárez Acosta – 2020038304

Guía de instalación y uso de la tarea

Antes de ejecutar cualquier comando se debe tener instalado en el equipo:

1. Docker desktop con kubernetes habilitado
2. Helm
3. Powershell
4. Lens

1.Desde línea de comandos, se ejecutan:

```
helm install monitoring monitoring --dependency-build
```

2.Luego descargar todas las dependencias de las herramientas a utilizar

```
helm install monitoring monitoring --dry-run
```

2.Actualizamos las dependencias:

```
helm install database database --dependency-build
```

3.Para instalar las bases de datos se ejecuta el siguiente comando:

```
helm install database database --dry-run
```

Configuración de Grafana

Una vez teniendo instaladas las bases de datos que se van a utilizar y las herramientas para monitorearlas, proseguimos con Grafana.

Para su funcionamiento, es necesario dirigirse a la carpeta de grafana-config, y luego de esa a la de dashboards. Luego, en el archivo value.yaml se deben agregar en el "file:" para cada base, el nombre del archivo de su respectivo dashboard.

Añadiendo los dashboard de las bases de datos

Para este paso vamos a buscar en internet "Grafana Dashboards" y visitaremos el primer sitio. En el buscador de dicho sitio, se va a proceder a buscar los dashboards respectivos a cada base de datos que se quiere monitorear.

Después de un proceso de selección, y posterior a haber realizado la descarga de cada dashboard, es necesario guardar estos archivos en una ruta específica.

Esta ruta corresponde a la carpeta donde se encuentre la tarea, siguiendo esta dirección: TareaCorta1 -> charts -> grafana-config -> dashboards.

Y en ese punto se guardan todos los archivos .json correspondientes a los dashboards.

Posterior a esto, en los pods se busca Grafana, y se hace un post forward. Una vez que se despliegue la ventana correspondiente, se debe poner el check de abrir en el browser.

En este punto se solicitará un usuario y una contraseña. El usuario corresponde a "admin", y la contraseña se encuentra en secrets de Grafana.

Una vez después de ingresar, nos dirigimos a dashboards y monitoring, y aquí se podrán visualizar los dashboards añadidos. De esta forma se puede ingresar a los dashboards.

Configuración de las herramientas

Kubernetes (kubectl)

Instalación

Para instalar Kubernetes en Windows, primero se debe ejecutar el siguiente comando en Powershell:

```
curl.exe -LO "https://dl.k8s.io/release/v1.26.0/bin/windows/amd64/kubectl.exe"
```

Una vez instalado, se debe agregar el Path a las System variables como C:\kube

Existe la alternativa de, si se tiene Docker instalado, se puede habilitar la opción "Enable Kubernetes" en las configuraciones de Docker Desktop y esto instalará Kubernetes automáticamente.

Uso

El primer paso, una vez que se tiene instalado kubectl, es crear un cluster, para esto se recomienda utilizar la herramienta Minikube el cual es un sistema gratuito que realiza algunas de las configuraciones necesarias para su uso, además de ofrecer herramientas para visualizar el cluster y sus componentes. Al tener un cluster creado, existen diferentes comandos útiles que permiten la creación, visualización y edición de deployments (grupos de pods con la misma configuración) por medio de Servicios de Kubernetes y pods (pueden contener a uno o más contenedores).

MariaDB

Instalación

MariaDB se puede instalar desde la página oficial, llamada MariaDB Foundation. En el proceso instalación, se necesita crear una contraseña con la que se iniciará sesión como root user en la base de datos, para el resto de configuraciones no hace falta realizar cambios y se puede instalar con los defaults.

Una vez hecha la instalación, se puede conectar a la base de datos ingresando desde la terminal a la carpeta *bin* ejecutando el comando:

```
mysql -u root -p
```

Seguido por la contraseña creada previamente.

MongoDB

En primera instancia se debe habilitar el puerto 20017 en el firewall. Posteriormente, a través de la consola SSH a tu servidor y verifica la versión de MongoDB con el comando:

```
mongod --version
```

A continuación, se debe crear un usuario administrador. Para ellos, se ejecuta una instancia de MongoShell y se hace en el puerto en el cual está configurado, esto se hace de la siguiente forma:

```
mongo --port 20017

use admin

db.createUser(
  {
    user: "myServerAdmin",
    pwd: "mipassword",
    roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
  }
);
```

Esto se hace cambiando los datos de "user" y "pwd" del código con el usuario y contraseña utilizadas en la tarea.

De esta forma ya queda configurada la herramienta de MongoDB.

Elasticsearch

Primeramente se deben haber instalado una versión de JDK compatible con Elasticsearch, y también se debe haber descargado la versión certificada de RPM de Elasticsearch.

Para instalar Elasticsearch se ejecuta el comando:

```
rpm -i elasticsearch-<version>.rpm
```

También se debe verificar que el usuario de Elasticsearch tenga acceso a Java mediante el comando:

```
sudo -u elasticsearch java -version
```

Posterior a esto, se actualiza el tamaño de la pila de Elasticsearch predeterminado realizando una modificación de la propiedad ES_HEAP_SIZE en el archivo /etc/sysconfig/elasticsearch.

Para finalizar se reinicia Elasticsearch mediante el comando:

```
/etc/init.d/elasticsearch restart
```

PostgreSQL

En primera instancia se debe estar seguro de que esté configurado para iniciar sesión desde localhost. Para esto se abre el archivo pg_hba.conf ubicado en el directorio de configuración y se modifica de esta forma:

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# "local" is for Unix domain socket connections only					
local	all	all			peer
# IPv4 local connections:					
host	all	all	127.0.0.1/32		md5
# IPv6 local connections:					
host	all	all	::1/128		md5

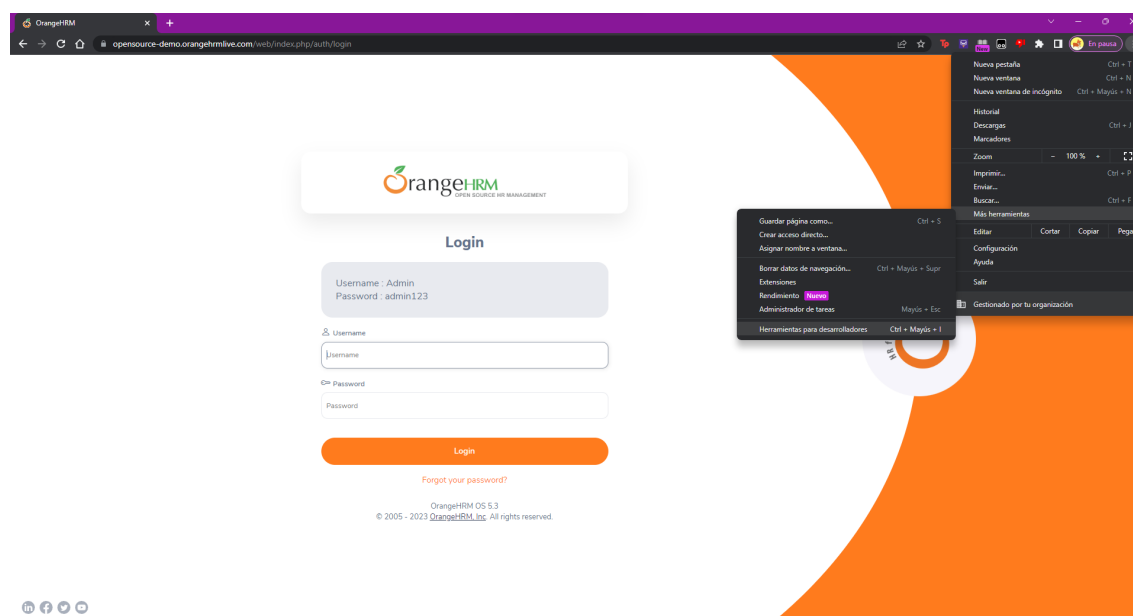
Pruebas de Gatling

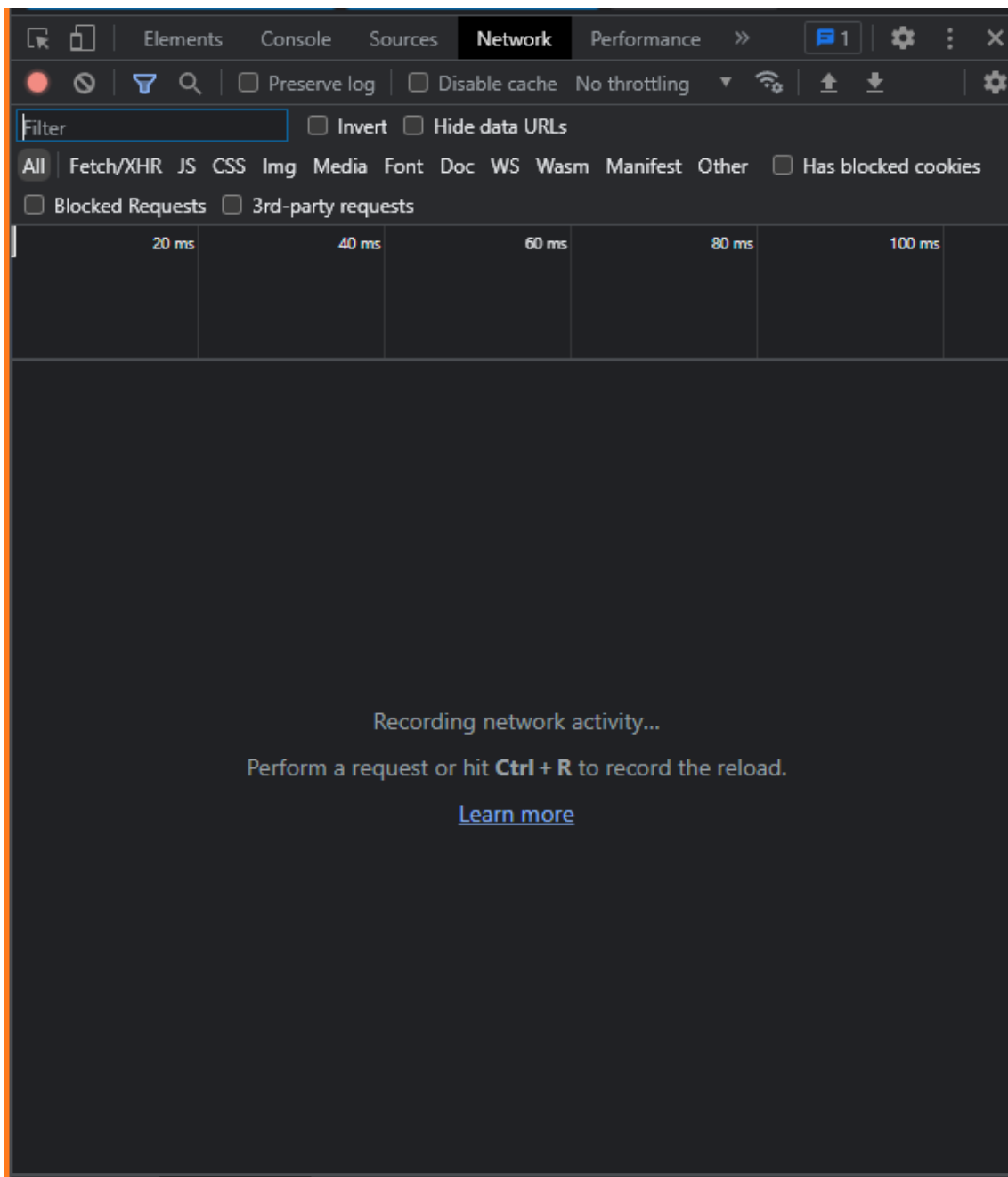
Para realizar las pruebas en Gatling se intentó inicialmente utilizar Flask como intermediario entre las bases de datos y Gatling. Se siguió la guía de la documentación del sitio web oficial de Flask (Flask, 2010) para montar la aplicación de Flask. Se logró montar la aplicación de Flask con normalidad y siguiendo la guía fue muy sencillo, sin embargo a varios integrantes el módulo de Flask aparecía sin instalar a pesar de haber hecho el "pip install".

Al no poder utilizar Flask como parte de la implementación de gatling se intentó usar Gatling directamente descargando el Gatling Bundle de su página oficial. Se siguió la guía del video publicado por Automata Step By Step (Pal, 2022). Antes de empezar al proyecto de Gatling se necesita lo siguiente:

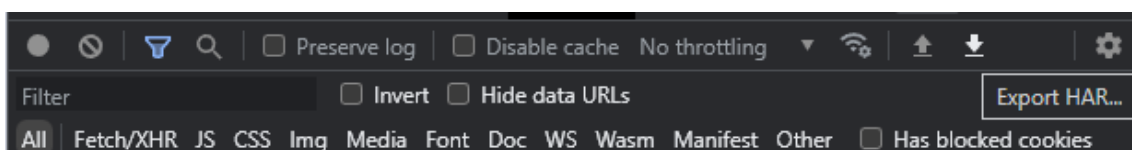
- Tener instalado el JDK.
- Descargar el Gatling Bundle
- Instalar el lenguaje de programación Scala.
- Descargar Maven para crear el proyecto.





A la hora de realizar pruebas de gatling por si solo, se utilizó el sitio web demo de Orange HRM (<https://opensource-demo.orangehrmlive.com/web/index.php/auth/login>). Al estar dentro de este sitio web, se presiona F12 o se abren las herramientas de desarrollador y se va a la sección que dice "Red" o "Network". En la sección de Red se presiona el botón de grabar y se comienzan a ingresar datos en la página demo.



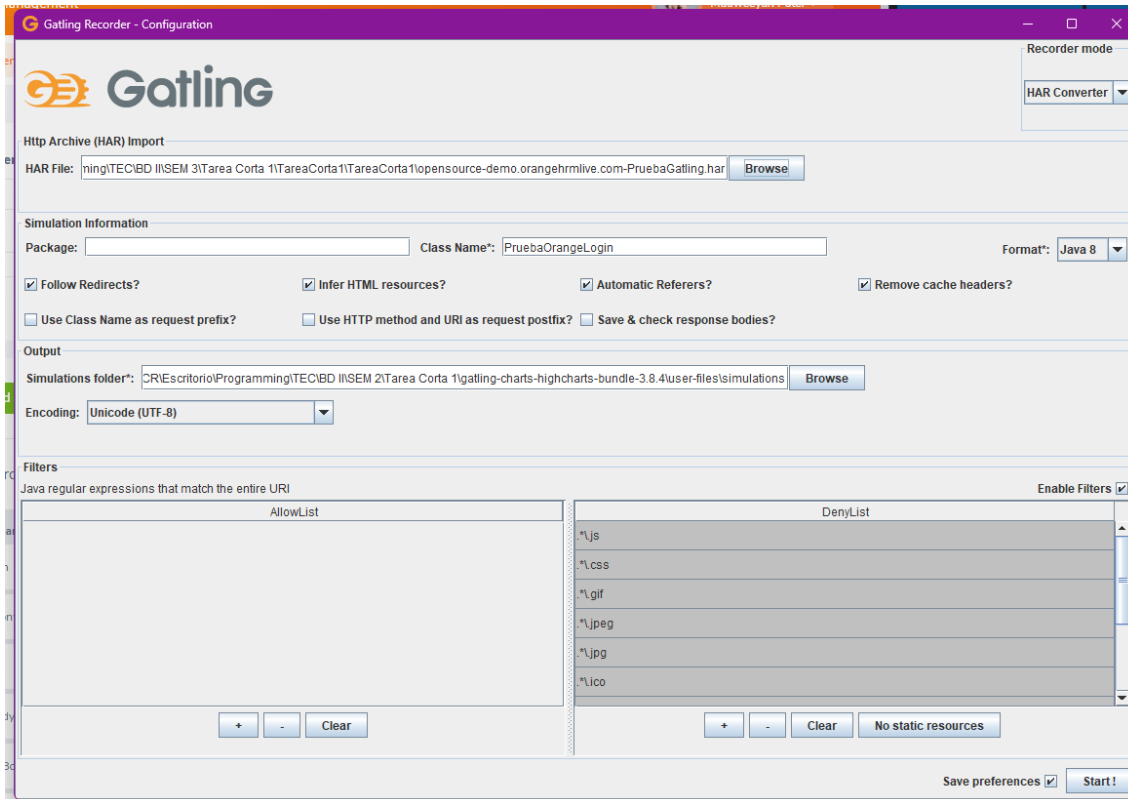


Después de grabar las acciones, se selecciona la opción de guardar los datos como un archivo .HAR y este se ingresa a la interfaz que se despliega al correr el archivo recorder.bat del Gatling Bundle.



 gatling	9/18/2022 11:22 AM	Windows Batch File	3 KB
 gatling	9/18/2022 11:22 AM	Shell Script	2 KB
 recorder	9/18/2022 11:22 AM	Windows Batch File	3 KB
 recorder	9/18/2022 11:22 AM	Shell Script	2 KB

Al ingresar el archivo gatling genera un link en el que se pueden observar los gráficos de las pruebas realizadas, además de crear un script para correr las pruebas el cual se puede editar también para alterar la cantidad de usuarios que "ingresarían a la base" en la simulación de Gatling. En este caso haremos las pruebas simulando 10 usuarios simultáneamente.



Gatling Recorder - Configuration

Http Archive (HAR) Import

HAR File:

Simulation Information

Package: Class Name: Format:

☒ Follow Redirects?
 ☒ Infer HTML resources?
 ☒ Automatic Referers?
 ☒ Remove cache headers?

☐ Use Class Name as request prefix?
 ☐ Use HTTP method and URI as request postfix?
 ☐ Save & check response bodies?

Output

Simulations folder*:

Encoding:

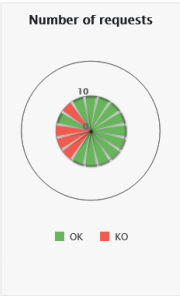
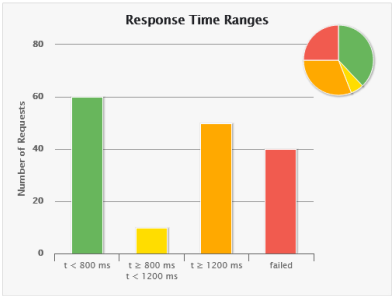
Filters

Java regular expressions that match the entire URI

☒ Enable Filters

☒ Save preferences

La prueba con el sitio web demo y sin conecciones se realizaron con éxito, sin embargo no se logró enlazar Gatling directamente con los pods de las bases de datos. Estos fueron los reportes generados:



Gatting Version
Version: 3.8.4
Released: 2022-09-13

Run Information
Date: 2023-03-27 05:30:58 GMT
Duration: 7s
Description: —

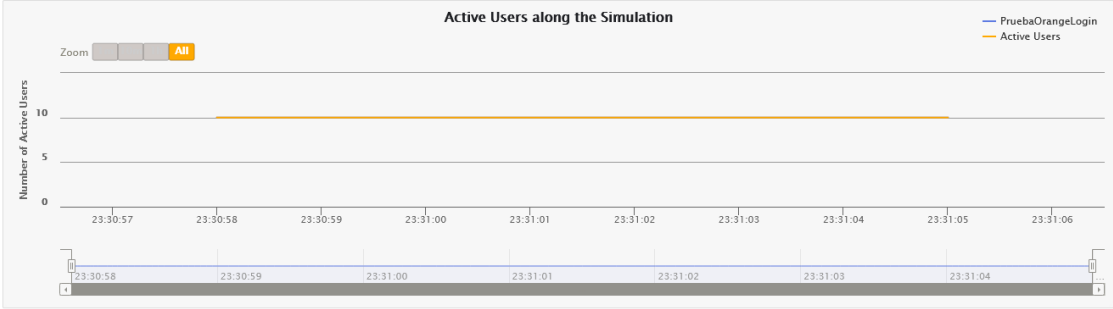
Expand all groupsCollapse all groups

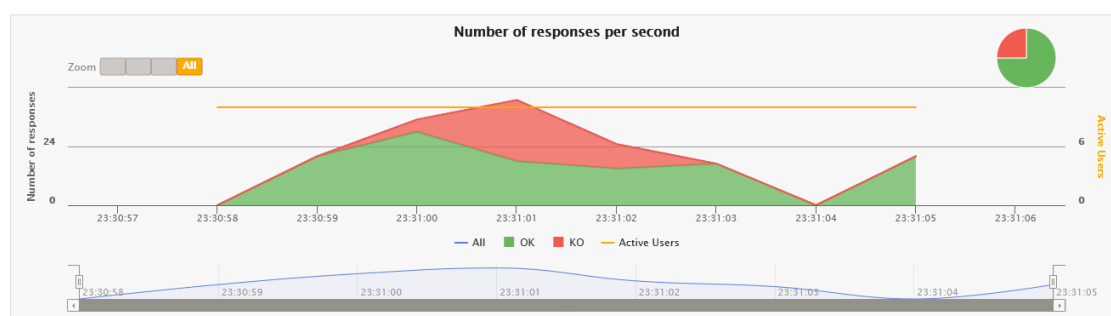
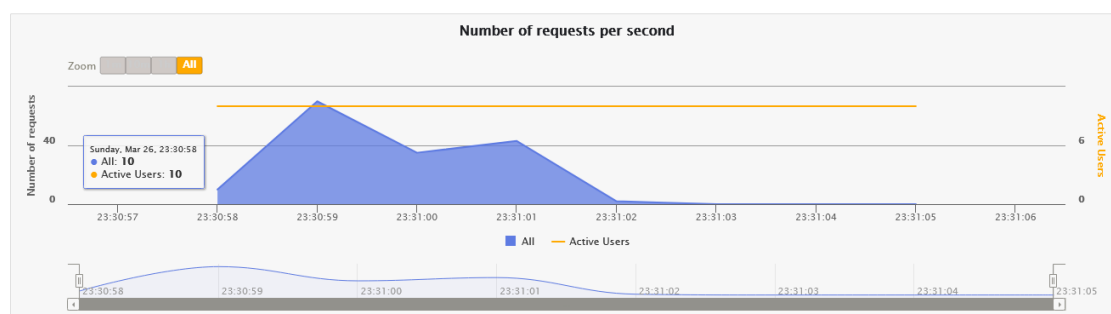
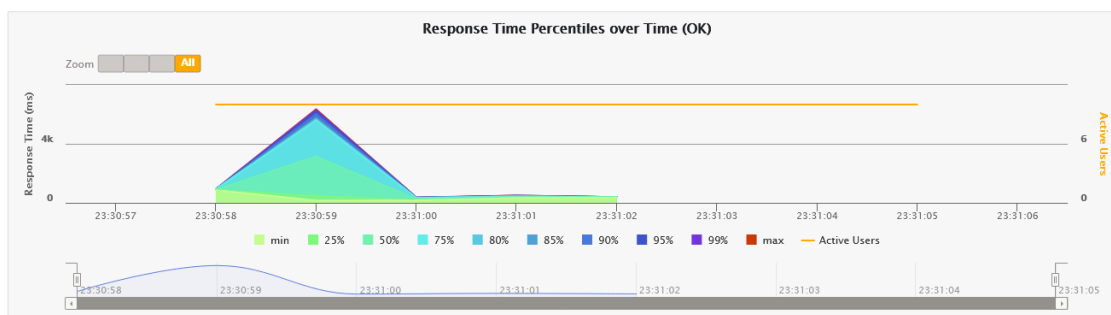
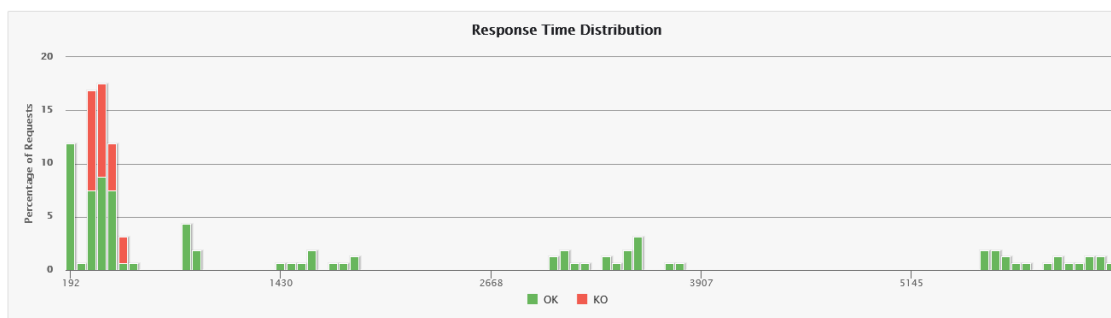
Requests +	Executions				Cnt/s +	Response Time (ms)								Mean +	Std.Dev +
	Total +	OK +	KO +	% KO +		Min +	50th pct +	75th pct +	95th pct +	99th pct +	Max +				
All Requests	160	120	40	25%	20	161	419	2170	5982	6242	6352			1533	1933
request_0	10	10	0	0%	1.25	857	892	920	932	932	932			896	26
request_0 Redirect 1	10	10	0	0%	1.25	335	421	456	507	533	539			421	59
favicon.ico?1672659722816	10	10	0	0%	1.25	161	174	178	189	192	193			175	9
chunk-vendors.css?1672659722816	10	10	0	0%	1.25	3011	3136	3413	3501	3509	3511			3221	187
app.css?1672659722816	10	10	0	0%	1.25	3335	3533	3561	3757	3763	3765			3528	138
chunk-vendors.js?1672659722816	10	10	0	0%	1.25	5644	5996	6216	6310	6344	6352			5979	255
app.js?1672659722816	10	10	0	0%	1.25	5563	5734	6046	6148	6167	6172			5816	230
request_1	10	10	0	0%	1.25	1413	1617	1785	1864	1884	1889			1653	150
request_2	10	10	0	0%	1.25	317	334	348	359	361	361			337	14
request_3	10	10	0	0%	1.25	162	167	195	217	223	224			180	22
request_4	10	0	10	100%	1.25	378	436	487	500	501	501			436	50

request_5	10	0	10	100%	1.25	315	322	337	355	361	362	329	15
request_6	10	0	10	100%	1.25	317	345	353	391	403	406	345	27
request_7	10	10	0	0%	1.25	324	348	405	440	450	452	371	43
request_8	10	0	10	100%	1.25	366	406	419	475	508	516	413	38
request_7 Redirect 1	10	10	0	0%	1.25	357	424	438	485	495	498	426	37

Errors

Error +	Count +	Percentage +
status.find.in([200, 209], 304), found 401	40	100 %





Conclusiones

Esta tarea fue de vital importancia para el aprendizaje y práctica de la utilización de herramientas como Docker Desktop, Kubernetes, Lens, Helm, Grafana, Gatling, etc.

Fue muy enriquecedor trabajar en la realización de la API hecha en Python con Flask, ya que es de conocimiento general que este tipo de aplicaciones intermediarias serán creadas y utilizadas durante el resto del curso y en los proyectos futuros.

Recomendaciones

- Es importante tener el "enable" en true de únicamente la base de datos que se va a monitorear, ya que si todas están activas al mismo tiempo, pueden consumir demasiados recursos del equipo.
- Es recomendable revisar que el proceso "Vmmem" asociado a Dockers tenga un consumo de memoria y procesador limitados, ya que en varios miembros del grupo representó una dificultad para poder utilizar el equipo, debido a un alto consumo en el CPU y la memoria. Para esto se encontró gran variedad de soluciones propuestas en la plataforma de YouTube.

Referencias

YouTube. (2022). Gatling Beginner Tutorial 1 | Load Testing, Introduction, Download, Setup |. YouTube. Recuperado 24 de marzo, 2023, de <https://www.youtube.com/watch?v=CPBWawzVeTo&t=6s>.

Loewen, C. (2022, 12 agosto). Advanced settings configuration in WSL. Microsoft Learn. Recuperado 24 de marzo de 2023, de <https://learn.microsoft.com/en-us/windows/wsl/wsl-config#configure-global-options-with-wslconfig>

YouTube. (2022). Proceso Vmmem de Docker en Windows consume mucha RAM (SOLUCIÓN) |. YouTube. Recuperado 24 de marzo de 2023, de <https://www.youtube.com/watch?v=CPBWawzVeTo&t=6s>.

Flask. (2010). Application setup. Application Setup - Flask Documentation (2.2.x). Recuperado 25 de marzo, 2023, de <https://flask.palletsprojects.com/en/2.2.x/tutorial/factory/>