**Instructivo de Instalación y Despliegue del Proyecto**

1. **Introducción**

Este instructivo detalla el proceso para construir, contenerizar, publicar, y desplegar una arquitectura de microservicios compuesta por:

* Oracle Database como base de datos para cada microservicio
* ms-eventos (gestión de eventos)
* ms-participantes (registro y validación de inscripciones y participantes)
* ms-gateway (API Gateway con Spring Cloud Gateway)

El despliegue se realiza primero en entorno local mediante Docker Compose y luego en un clúster Kubernetes usando manifiestos YAML.

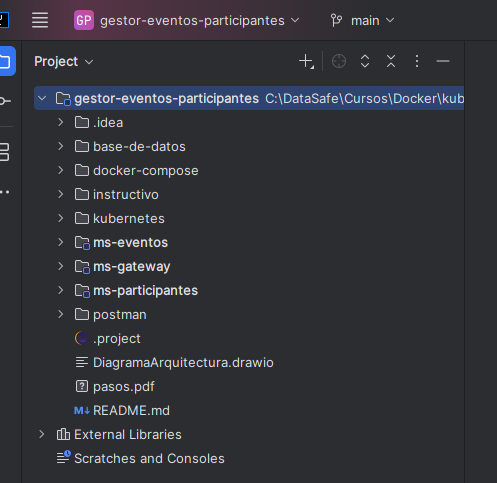
Clonamos el repositorio del proyecto:

git clone <https://github.com/rmcabrera/gestor-eventos-participantes.git>

1. **Construcción de los Microservicios**

#### Estructura general:

* ms-eventos: CRUD de eventos
* ms-participantes: validación de inscripciones conectándose a ms-eventos
* ms-gateway: punto de entrada que enruta hacia los otros servicios
* kubernetes: archivos yaml para despliegue en kubernetes
* Docker-compose: archivo para ejecución en Docker

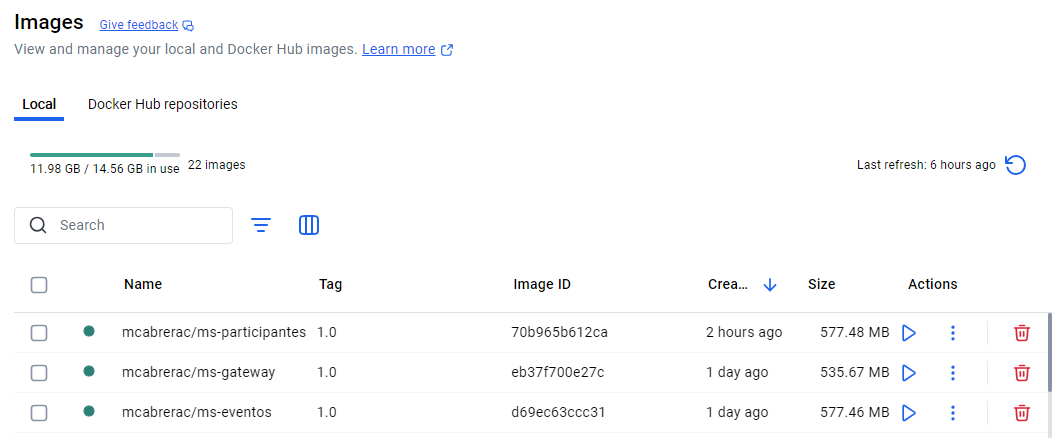


1. **Construcción de las Imágenes Docker**

docker build -t <usuario-dockerhub>/ms-eventos:1.0 .

docker build -t <usuario-dockerhub>/ms-participantes:1.0 .

docker build -t <usuario-dockerhub>/ms-gateway:1.0 .



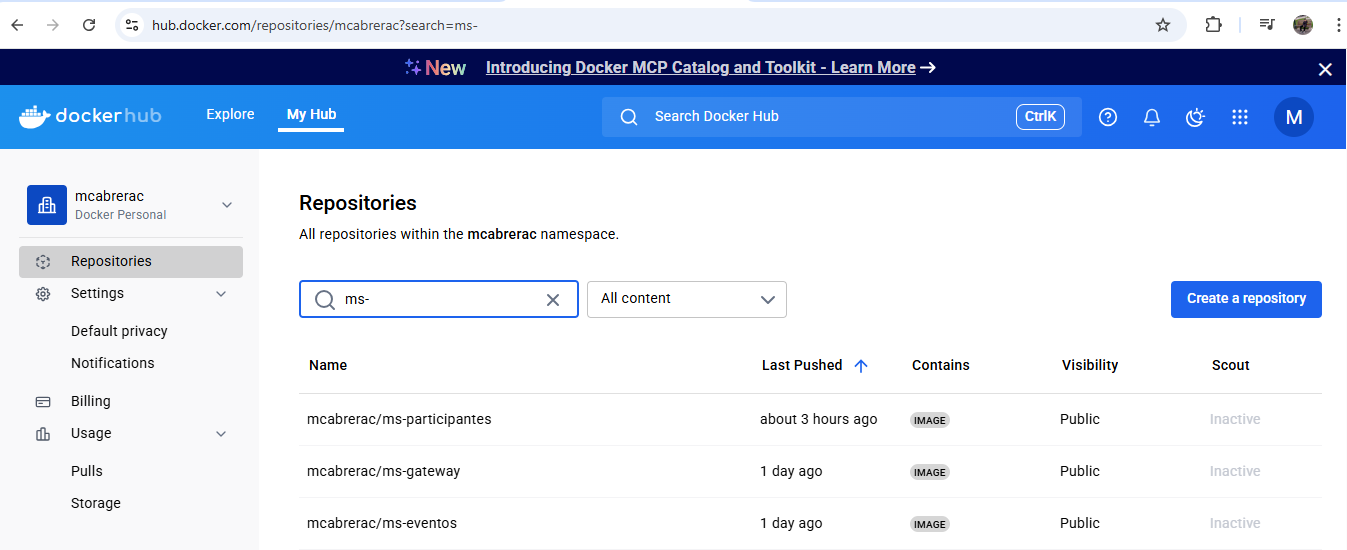
1. **Publicación en Docker Hub**

docker login

docker push <usuario-dockerhub>/ms-eventos:1.0

docker push <usuario-dockerhub>/ms-participantes:1.0

docker push <usuario-dockerhub>/ms-gateway:1.0



1. **Ejecución de Docker Compose**

Ingresar a la carpeta donde se encuentra el archivo docker-compose.yml

cd docker-compose

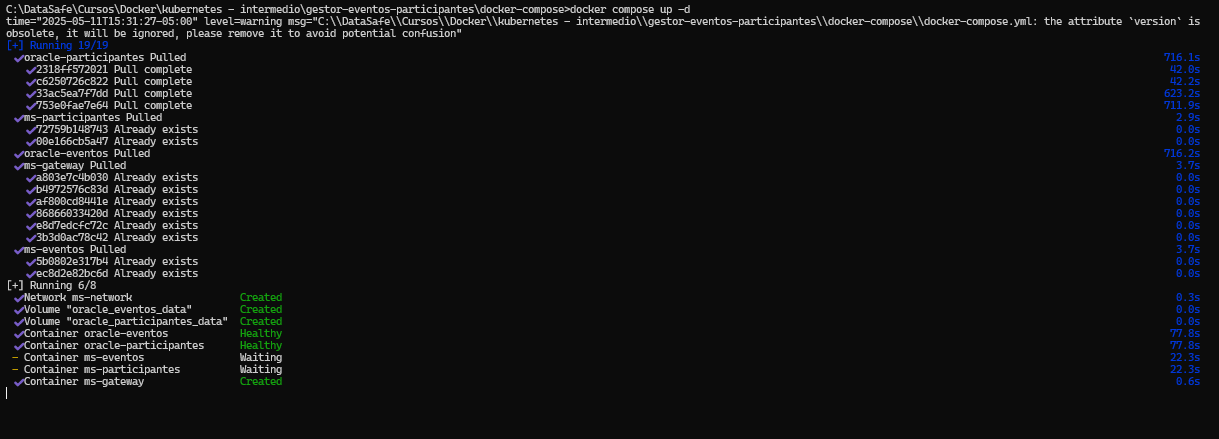
Podemos descargar la imagen de Oracle previamente para que la ejecución del docker compose sea más rápida:

docker pull container-registry.oracle.com/database/express:21.3.0-xe

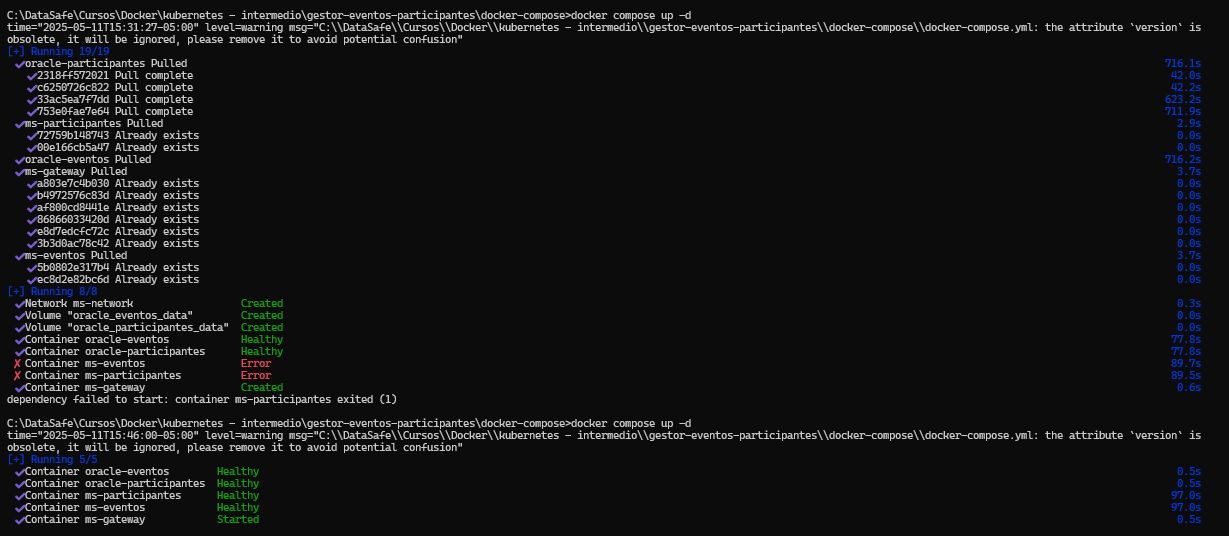
Caso contrario podemos ejecutar el comando para desplegar en Docker, el cual realizará la descarga de la imagen de oracle:

docker compose up -d





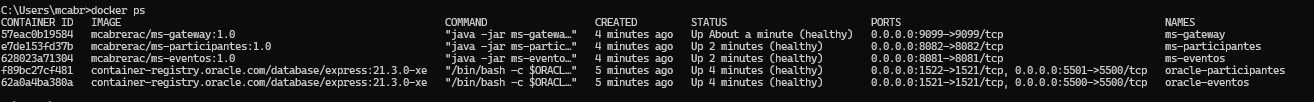
En caso de presentarse algún error, como se muestra en la siguiente imagen, solo volvemos a ejecutar el comando **docker compose up -d**



Ejecutamos lo siguiente:

docker compose ps

para validar que todo este conforme.

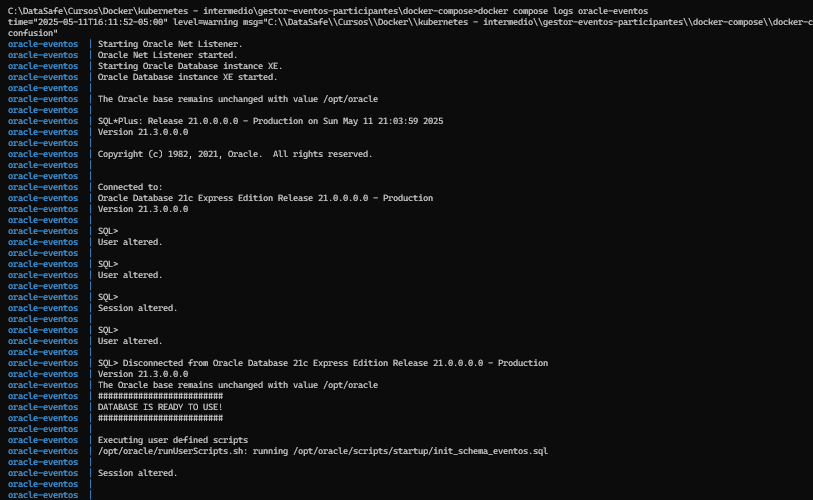


1. **Verificación de logs en Docker**

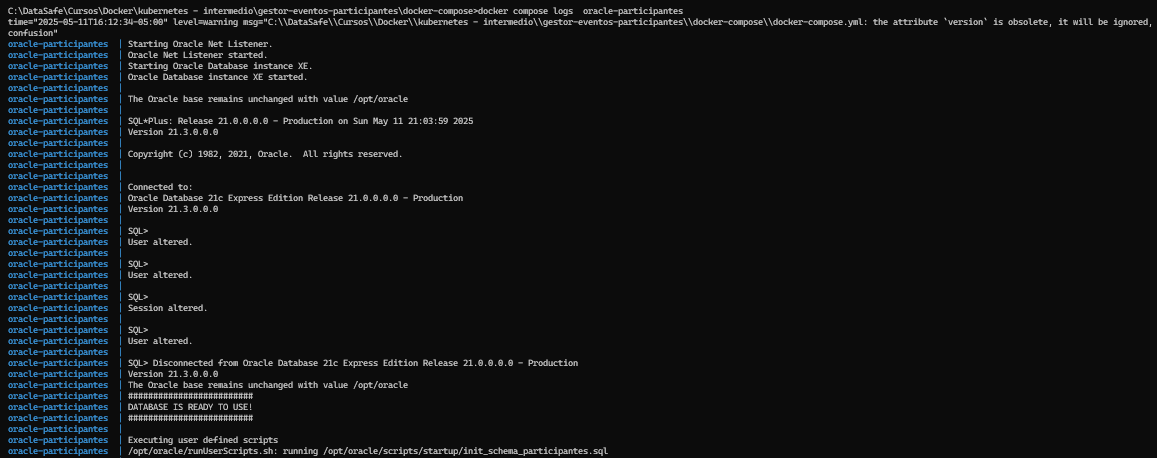
Para la verificación de los logs ejecutamos los siguientes comandos:

Con el resultado de cada log, podemos validar que las bases de datos, y los microservicios están ejecutándose correctamente.

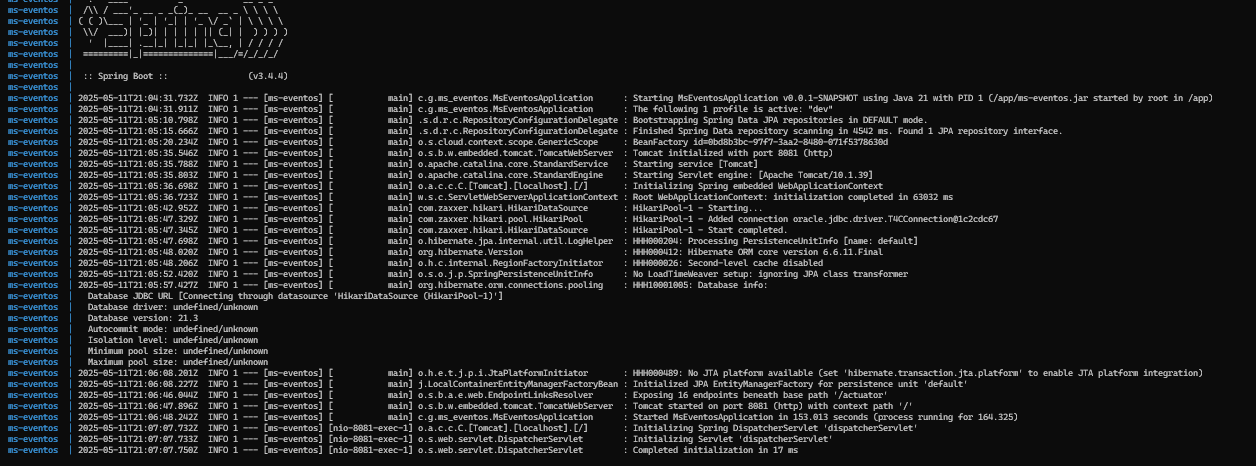
docker compose logs -f oracle-eventos



docker compose logs -f oracle-participantes



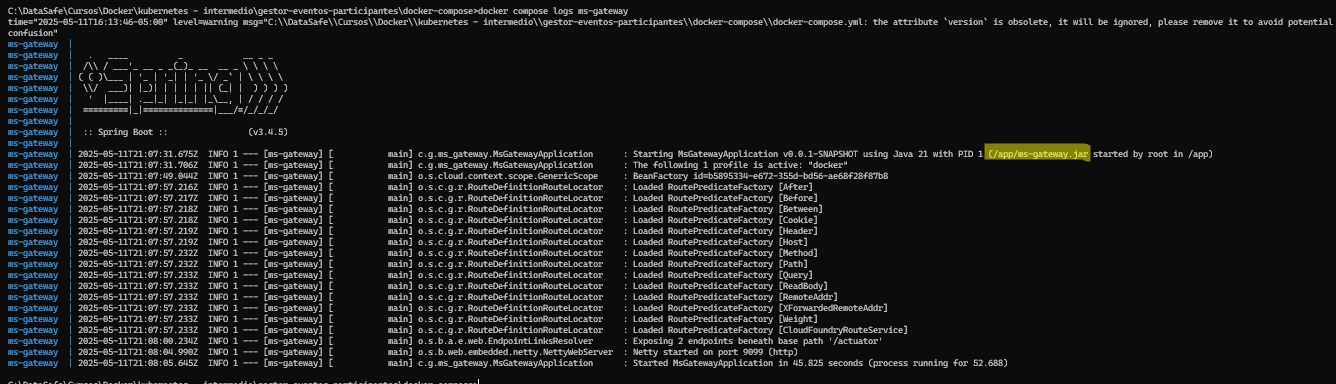
docker compose logs -f ms-eventos



docker compose logs -f ms-participantes



docker compose logs -f ms-gateway



1. **Pruebas de creación de eventos e inscripción de participantes.**

Las pruebas las podemos realizar mediante curl o postman.

***Pruebas con CURL***

curl -X POST -H "Content-Type: application/json; charset=UTF-8" \

-d '{

"nombre": "Conferencia Tech 2025",

"descripcion": "Un evento sobre las \u00faltimas tecnolog\u00edas III",

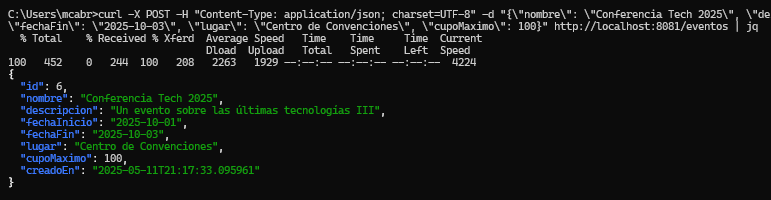
"fechaInicio": "2025-10-01",

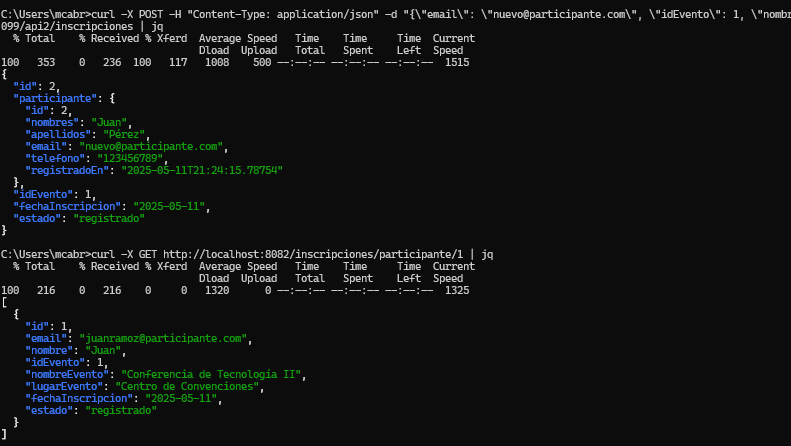
"fechaFin": "2025-10-03",

"lugar": "Centro de Convenciones",

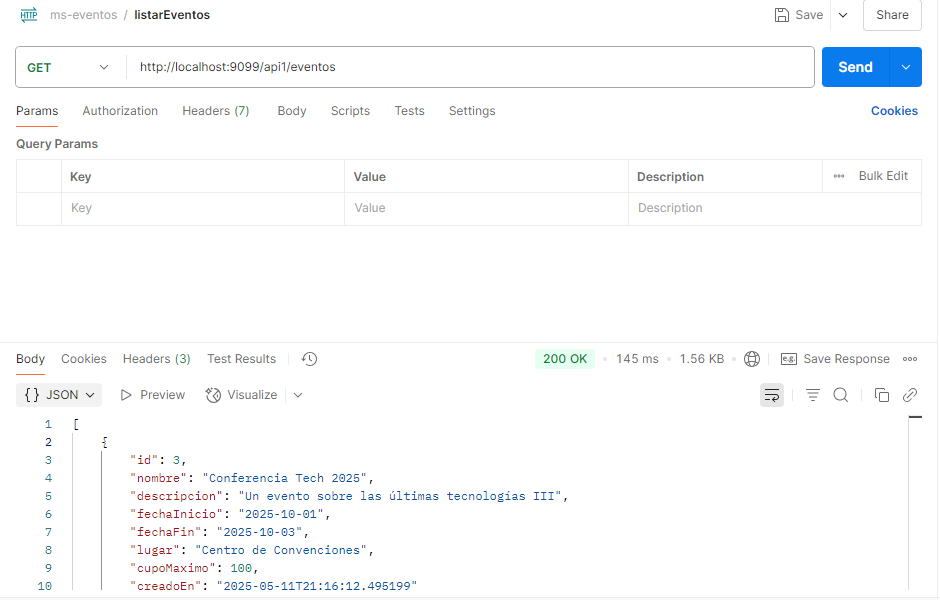
"cupoMaximo": 100

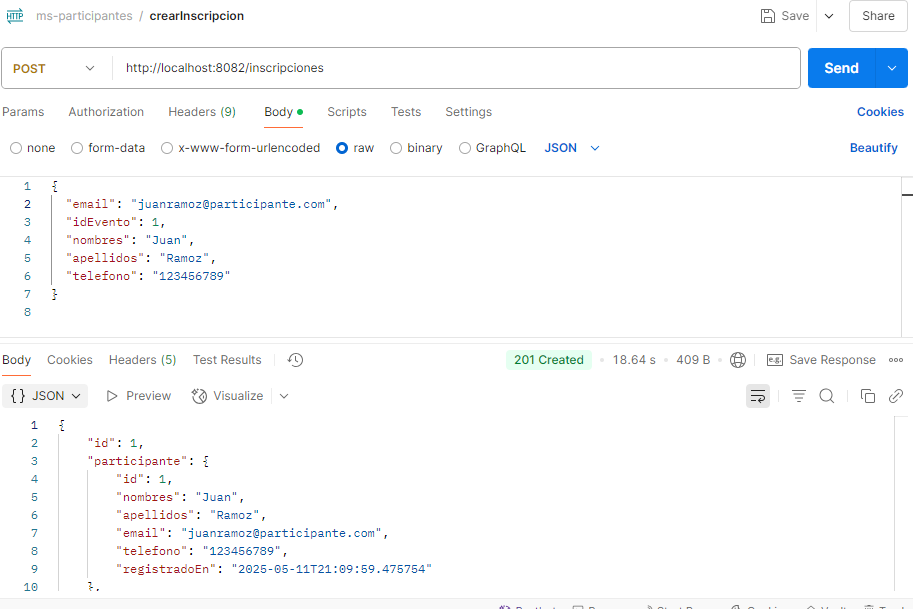
}' <http://localhost:8081/eventos>





***Pruebas con Postman***



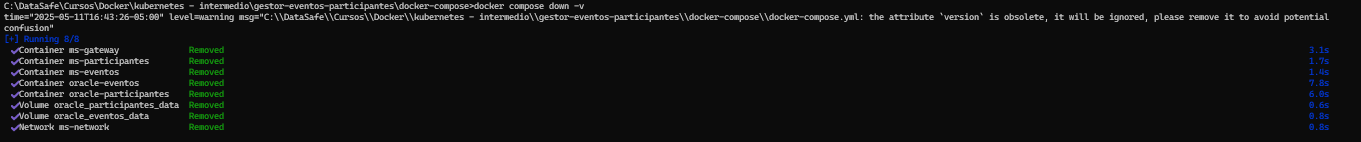


***Listado de métodos***

|  |  |  |  |
| --- | --- | --- | --- |
| **Microservicio** | **Método HTTP** | **URL** | **Ejemplo** |
| ms-eventos | POST | <http://localhost:8081/eventos> | curl -X POST -H "Content-Type: application/json; charset=UTF-8" \  -d '{  "nombre": "Conferencia Tech 2025",  "descripcion": "Un evento sobre las \u00faltimas tecnolog\u00edas III",  "fechaInicio": "2025-10-01",  "fechaFin": "2025-10-03",  "lugar": "Centro de Convenciones",  "cupoMaximo": 100  }' http://localhost:8081/eventos |
| ms-gateway | POST | <http://localhost:9099/api1/eventos> | curl -X POST -H "Content-Type: application/json; charset=UTF-8" \  -d '{  "nombre": "Conferencia Tech 2025",  "descripcion": "Un evento sobre las \u00faltimas tecnolog\u00edas III",  "fechaInicio": "2025-10-01",  "fechaFin": "2025-10-03",  "lugar": "Centro de Convenciones",  "cupoMaximo": 100  }' http://localhost:9099/api1/eventos |
| ms-eventos | GET | <http://localhost:8081/eventos> | curl -X GET http://localhost:8081/eventos |
| ms-gateway | GET | <http://localhost:9099/api1/eventos> | curl -X GET http://localhost:9099/api1/eventos |
| ms-eventos | GET | [http://localhost:8081/eventos/{id}](http://localhost:8081/eventos/%7Bid%7D) | curl -X GET http://localhost:8081/eventos/1 |
| ms-eventos | PUT | [http://localhost:8081/eventos/{id}](http://localhost:8081/eventos/%7Bid%7D) | curl -X PUT -H "Content-Type: application/json; charset=UTF-8" -d "{\"nombre\": \"Conferencia Tech 2025\", \"descripcion\": \"Un evento sobre las \u00faltimas tecnolog\u00edas III\", \"fechaInicio\": \"2025-10-01\", \"fechaFin\": \"2025-10-03\", \"lugar\": \"Centro de Convenciones\", \"cupoMaximo\": 100}" http://localhost:8081/eventos/1 |
| ms-eventos | DELETE | [http://localhost:8081/eventos/{id}](http://localhost:8081/eventos/%7Bid%7D) | curl -X DELETE http://localhost:8081/eventos/{id} |
| ms-participantes | POST | <http://localhost:8082/inscripciones> | curl -X POST -H "Content-Type: application/json" -d "{\"email\": \"robertoc@participante.com\", \"idEvento\": 1, \"nombres\": \"Roberto\", \"apellidos\": \"Cabrera\", \"telefono\": \"123456789\"}" http://localhost:8082/inscripciones |
| ms-gateway | POST | <http://localhost:9099/api2/inscripciones> | curl -X POST -H "Content-Type: application/json" -d "{\"email\": \"raul@participante.com\", \"idEvento\": 1, \"nombres\": \"Raul\", \"apellidos\": \"Arrascue\", \"telefono\": \"123456789\"}" http://localhost:9099/api2/inscripciones |
| ms-participantes | GET | [http://localhost:8082/inscripciones/participante/{id}](http://localhost:8082/inscripciones/participante/%7Bid%7D) | curl -X GET http://localhost:8082/inscripciones/participante/1 |
| ms-gateway | GET | [http://localhost:9099/api2/inscripciones/participante/{id}](http://localhost:9099/api2/inscripciones/participante/%7Bid%7D) | curl -X GET http://localhost:9099/api2/inscripciones/participante/1 |

1. **Detener y limpiar Docker compose**

docker compose down -v



1. **Despliegue en Kubernetes**

Para desplegar y ejecutar los archivos para kubernetes, regresamos a la ruta principal,

cd..

**Verificar imagen de Oracle**

Solo ejecutar en caso de no existir la imagen de Oracle descargada

docker pull container-registry.oracle.com/database/express:21.3.0-xe

Ejecutamos los siguientes comandos:

***Crear Namespace***

kubectl apply -f kubernetes/kb-namespace/namespace-gestion-ep-dev.yaml

***Crear Persistent Volumes***

kubectl apply -f kubernetes/kb-eventos/pv-db-eventos.yaml

kubectl apply -f kubernetes/kb-participantes/pv-db-participantes.yaml

***ConfigMaps (inicialización y configuración)***

kubectl apply -f kubernetes/kb-eventos/configmap-db-eventos.yaml

kubectl apply -f kubernetes/kb-participantes/configmap-db-participantes.yaml

kubectl apply -f kubernetes/kb-microservicios/configmap-ms-eventos.yaml

kubectl apply -f kubernetes/kb-microservicios/configmap-ms-participantes.yaml

***Secrets***

kubectl apply -f kubernetes/kb-microservicios/secret-db.yaml

***Roles y Permisos***

kubectl apply -f kubernetes/kb-microservicios/role.yaml

kubectl apply -f kubernetes/kb-microservicios/rolebinding.yaml

kubectl apply -f kubernetes/kb-microservicios/cluster-role.yaml

kubectl apply -f kubernetes/kb-microservicios/cluster-role-binding.yaml

***Persistent Volume Claims***

kubectl apply -f kubernetes/kb-eventos/pvc-db-eventos.yaml

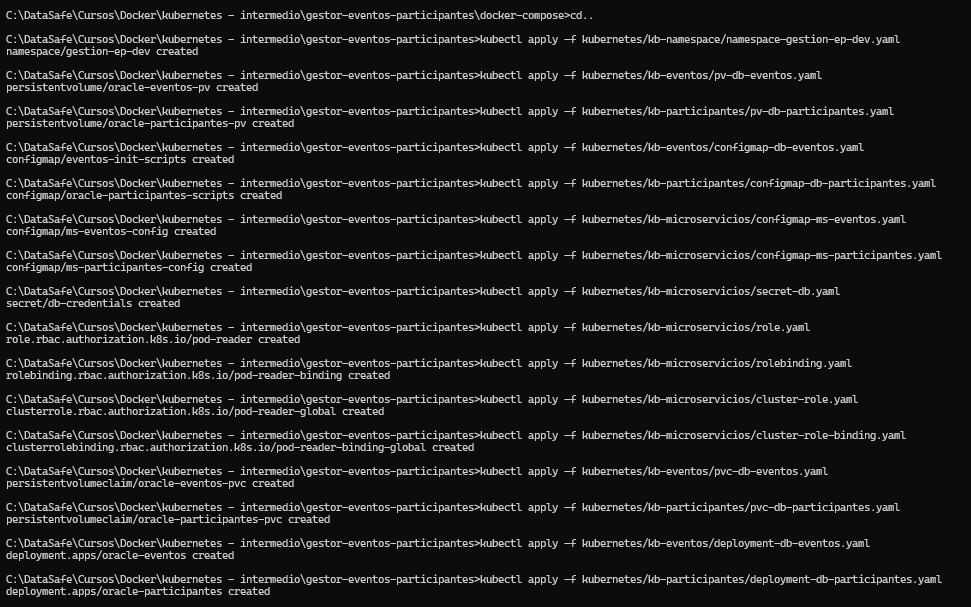
kubectl apply -f kubernetes/kb-participantes/pvc-db-participantes.yaml

***Deployments***

kubectl apply -f kubernetes/kb-eventos/deployment-db-eventos.yaml

kubectl apply -f kubernetes/kb-participantes/deployment-db-participantes.yaml

Hasta este punto podemos observar en la imagen la creación de todo lo ejecutado.



Para poder continuar con la ejecución de los siguientes comandos, debemos esperar que las bases de datos se hayan terminado de ejecutar, y así evitar errores al momento del despliegue:

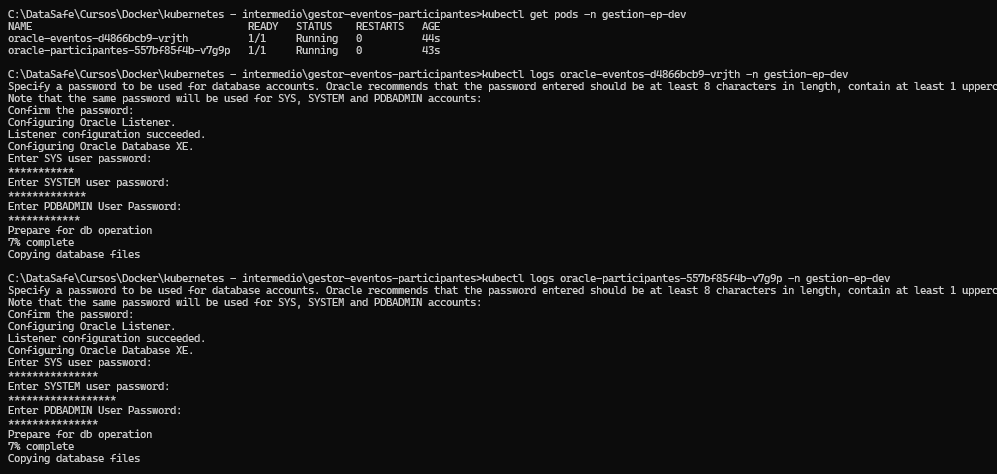
Verificamos los pods que se están ejecutando, para obtener sus nombres:

kubectl get pods -n gestion-ep-dev

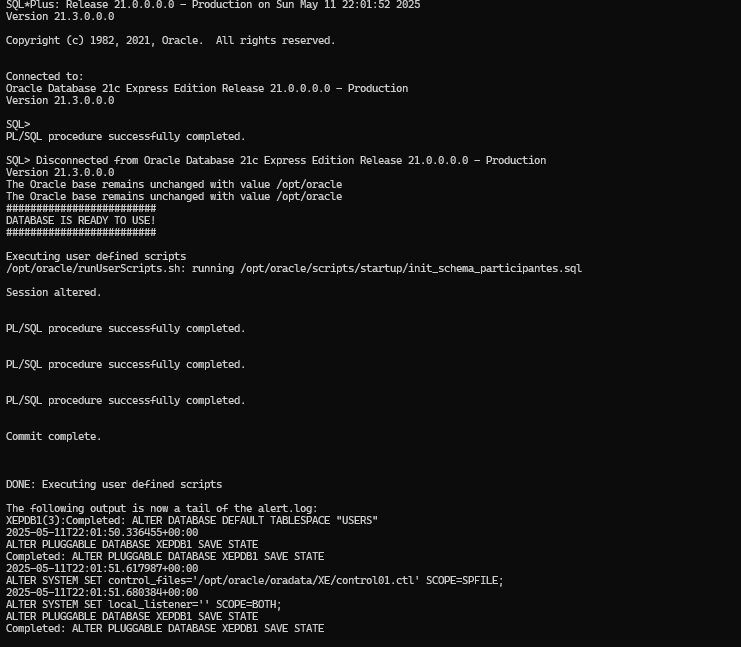
Verificamos el log de los pods que se están ejecutando:

kubectl logs <podName> -n gestion-ep-dev

Cuando ejecutamos para verificar el log de cada pod en ejecución podemos verificar que aún se están configurando las bases de datos, para ambos casos aú se encuentran al 7%, como se puede apreciar en las imágenes, debemos esperar que terminen para continuar con el proceso de ejecución.



Una vez terminada la creación de las bases de datos, se mostrará en log que el proceso se ha completado, como se muestra en la siguiente imagen:



Después de haber verificado que el proceso de creación de las bases de datos ha terminado, podemos ejecutar el despliegue de los demás archivos yaml.

kubectl apply -f kubernetes/kb-microservicios/deployment-ms-eventos.yaml

kubectl apply -f kubernetes/kb-microservicios/deployment-ms-participantes.yaml

kubectl apply -f kubernetes/kb-microservicios/deployment-ms-gateway.yaml

***Services***

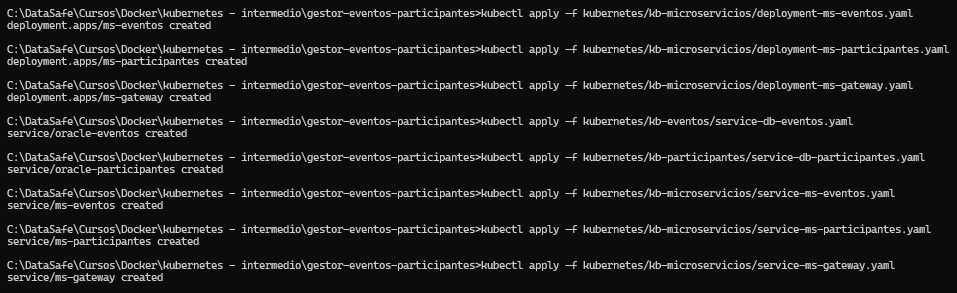
kubectl apply -f kubernetes/kb-eventos/service-db-eventos.yaml

kubectl apply -f kubernetes/kb-participantes/service-db-participantes.yaml

kubectl apply -f kubernetes/kb-microservicios/service-ms-eventos.yaml

kubectl apply -f kubernetes/kb-microservicios/service-ms-participantes.yaml

kubectl apply -f kubernetes/kb-microservicios/service-ms-gateway.yaml

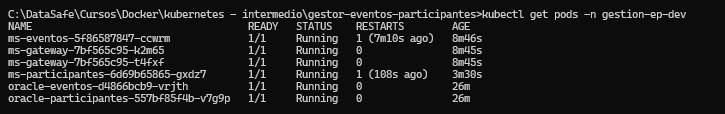


***Verificar***

kubectl get all -n gestion-ep-dev

Validamos que todos este en Running mediante el siguiente comando:

kubectl get pods -n gestion-ep-dev



1. **Pruebas de creación de eventos e inscripción de participantes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Microservicio** | **Método HTTP** | **URL** | **Ejemplo** |
| ms-eventos | POST | <http://localhost:30081/eventos> | curl -X POST -H "Content-Type: application/json; charset=UTF-8" \  -d '{  "nombre": "Conferencia Tech 2025",  "descripcion": "Un evento sobre las \u00faltimas tecnolog\u00edas III",  "fechaInicio": "2025-10-01",  "fechaFin": "2025-10-03",  "lugar": "Centro de Convenciones",  "cupoMaximo": 100  }' <http://localhost:30081/eventos> |
| ms-gateway | POST | <http://localhost:30099/api1/eventos> | curl -X POST -H "Content-Type: application/json; charset=UTF-8" \  -d '{  "nombre": "Conferencia Tech 2025",  "descripcion": "Un evento sobre las \u00faltimas tecnolog\u00edas III",  "fechaInicio": "2025-10-01",  "fechaFin": "2025-10-03",  "lugar": "Centro de Convenciones",  "cupoMaximo": 100  }' <http://localhost:30099/api1/eventos> |
| ms-eventos | GET | <http://localhost:30081/eventos> | curl -X GET http://localhost:30081/eventos |
| ms-gateway | GET | <http://localhost:30099/api1/eventos> | curl -X GET http://localhost:30099/api1/eventos |
| ms-eventos | GET | [http://localhost:30081/eventos/{id}](http://localhost:30081/eventos/%7bid%7d) | curl -X GET <http://localhost:30081/eventos/1> |
| ms-eventos | PUT | [http://localhost:30081/eventos/{id}](http://localhost:30081/eventos/%7bid%7d) | curl -X PUT -H "Content-Type: application/json; charset=UTF-8" -d "{\"nombre\": \"Conferencia Tech 2025\", \"descripcion\": \"Un evento sobre las \u00faltimas tecnolog\u00edas III\", \"fechaInicio\": \"2025-10-01\", \"fechaFin\": \"2025-10-03\", \"lugar\": \"Centro de Convenciones\", \"cupoMaximo\": 100}" <http://localhost:30081/eventos/1> |
| ms-eventos | DELETE | [http://localhost:30081/eventos/{id}](http://localhost:30081/eventos/%7bid%7d) | curl -X DELETE http://localhost:30081/eventos/{id} |
| ms-participantes | POST | <http://localhost:30082/inscripciones> | curl -X POST -H "Content-Type: application/json" -d "{\"email\": \"robertoc@participante.com\", \"idEvento\": 1, \"nombres\": \"Roberto\", \"apellidos\": \"Cabrera\", \"telefono\": \"123456789\"}" <http://localhost:30082/inscripciones> |
| ms-gateway | POST | <http://localhost:30099/api2/inscripciones> | curl -X POST -H "Content-Type: application/json" -d "{\"email\": \"raul@participante.com\", \"idEvento\": 1, \"nombres\": \"Raul\", \"apellidos\": \"Arrascue\", \"telefono\": \"123456789\"}" http://localhost:30099/api2/inscripciones |
| ms-participantes | GET | [http://localhost:30082/inscripciones/participante/{id}](http://localhost:30082/inscripciones/participante/%7bid%7d) | curl -X GET http://localhost:30082/inscripciones/participante/1 |
| ms-gateway | GET | [http://localhost:30099/api2/inscripciones/participante/{id}](http://localhost:30099/api2/inscripciones/participante/%7bid%7d) | curl -X GET <http://localhost:30099/api2/inscripciones/participante/1> |

1. **Limpiar Kubernetes**

kubectl delete namespace gestion-ep-dev

kubectl delete pv oracle-eventos-pv

kubectl delete pv oracle-participantes-pv

kubectl delete clusterrole pod-reader-global

kubectl delete clusterrolebinding pod-reader-binding-global

1. **Verificar recursos han sido eliminados**

kubectl get all -n gestion-ep-dev

1. **Apis con Swagger**

