

Práctica 1. Escalabilidad y tolerancia a fallos

Enunciado

El objetivo es crear una aplicación escalable y tolerante a fallos en Kubernetes. Partiendo de la aplicación [ejem4-webgatos2-scalable](#), se deberá modificar de la siguiente forma:

- Se usará Hazelcast en vez de MySQL para mantener la sesión sincronizada entre las réplicas. El test que verifica que la sesión se mantiene cuando se hacen peticiones a varias réplicas debería pasar.
- Pruebas de caos:
 - Se deberán ejecutar pruebas de carga con Artillery en varios escenarios diferentes y se compararán los resultados obtenidos. El objetivo es que se pueda ver cómo el uso de replicación y service Mesh reducen el número de errores obtenidos por los usuarios en caso de fallo.
 - Escenarios
 - Escenario 1: Un único pod en el deployment. No caos testing.
 - Escenario 2: Un único pod en el deployment. Caos testing con chaos-pod-monkey.
 - Escenario 3: Dos pods en el deployment. Caos testing con chaos-pod-monkey.
 - Escenario 4: Dos pods en el deployment. Uso de Gateway de Istio para aceptar peticiones. Caos testing con chaos-pod-monkey.

Se deberá entregar un vídeo mostrando la ejecución de las pruebas de carga con y sin istio

Formato de entrega

La práctica se entregará teniendo en cuenta los siguientes aspectos:

- La práctica se entregará como un fichero .zip con el código de los diferentes servicios y los ficheros auxiliares. El nombre del fichero .zip será el correo URJC del alumno (sin @alumnos.urjc.es).

Las prácticas se podrán realizar de forma individual o por parejas. En caso de que la práctica se haga por parejas:

- Sólo será entregada por uno de los alumnos
- El nombre del fichero .zip contendrá el correo de ambos alumnos separado por guión. Por ejemplo p.perezf2019-z.gonzalez2019.zip