Universidad de Guadalajara.

Computación Tolerante a fallas.

Diego Ivan Becerra Gonzalez.

Sección D06.

Objetivo.

Utilizar la herramienta cheeky monkey.

Desarrollo.

Para esta practica se retomo lo realizado en la practica de kubernetes, , que era crear una aplicación que simulara un API utilizando la librería express y un programa en JavaScript, solo que en vez de tener 3 replicas de un pod, se crearon 5 réplicas.

```
diego@LAPTOP-ECVOPM04 MINGW64 ~/OneDrive/Escritorio/PC/Sistemas tolerantes a fallas/practicas/
node-hello-app (cheekymonkey)
$ kubectl scale deployment node-app --replicas 5
deployment.apps/node-app scaled
```

```
C:\Users\diego>kubectl get pods
NAME                        READY   STATUS    RESTARTS   AGE
node-app-76b6cbdbd5-8qzxg   1/1     Running   0          6h40m
node-app-76b6cbdbd5-gfdd6   1/1     Running   0          11m
node-app-76b6cbdbd5-q58zf   1/1     Running   0          14m
node-app-76b6cbdbd5-tx9xk   1/1     Running   0          6h42m
node-app-76b6cbdbd5-vtslr   1/1     Running   0          6h40m
```

También se creo un deployment que se encargará de crear nuevos pods en caso de que alguno falle.

```
NAME                       READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/node-app   5/5     5            5           6h42m
```

Ahora sigue instalar cheeky monkey. Para esto se clono el repositorio de la herramienta en el link https://github.com/richstokes/cheekymonkey.

Después, nos vamos dentro de la carpeta en una terminal y ejecutamos el siguiente comando para instalar las dependencias necesarias: “pip install -r requirements.txt”.

Y finalmente ejecutamos “python cheekymonkey.py”.

Lo que hace el comando anterior es ejecutar la herramienta.

Lo que sigue es simplemente jugar con cheeky monkey. Al momento de destruir una caja, nos mostro que se destruyo un pod, como se puede observar en el siguiente log:

```
C:\Users\diego\Downloads\cheekymonkey>python cheekymonkey.py
INFO: Starting in online mode
INFO: Excluding namespaces:
INFO: Attempting to connect to Kubernetes API host..
INFO: Connected to Kubernetes host: https://kubernetes.docker.internal:6443
INFO: Number of pods: 14
INFO: Creating 14 crates
INFO: Updating pod list from: https://kubernetes.docker.internal:6443
INFO: Number of pods: 14
INFO: Killing Random pod: node-app-76b6cbdbd5-vtslr from namespace: default
```

Y al ejecutar el comando “kubectl get pods -w” obtenemos el estado de los pods, como muestra la siguiente imagen:

```
PS C:\Users\diego> kubectl get pods -w
NAME                        READY   STATUS        RESTARTS   AGE
node-app-76b6cbdbd5-8qzxg   1/1     Running       0          6h51m
node-app-76b6cbdbd5-bm8d8   1/1     Running       0          29s
node-app-76b6cbdbd5-gfdd6   1/1     Running       0          23m
node-app-76b6cbdbd5-q58zf   1/1     Running       0          25m
node-app-76b6cbdbd5-tx9xk   1/1     Running       0          6h54m
node-app-76b6cbdbd5-vtslr   1/1     Terminating   0          6h51m
node-app-76b6cbdbd5-vtslr   0/1     Terminating   0          6h51m
node-app-76b6cbdbd5-vtslr   0/1     Terminating   0          6h51m
node-app-76b6cbdbd5-vtslr   0/1     Terminating   0          6h51m
```

En la imagen anterior podemos observar como se destruyo un pod, pero también hay 5 pods corriendo, lo que significa que el deployment está haciendo su trabajo.

Conclusión.

Cheeky monkey es una herramienta curiosa y útil al momento de querer causar fallas aleatorias en nuestros pods, sin embargo, creo que su funcionalidad está muy limitada a solo destruir pods, cuando puede haber errores más complejos como mandar solicitudes incorrectas, o mandar demasiadas peticiones a un pod, además, considero que el causar fallas especificas es más útil que causar fallas aleatorias, porque con las fallas planeadas, sabes a ciencia cierta que estas probando.

Bibliografía.

Diego Ivan. (s. f.). *practicasSTA/Kubernetes.pdf at kubernetes · diegoivan1987/practicasSTA*. GitHub. https://github.com/diegoivan1987/practicasSTA/blob/kubernetes/Kubernetes.pdf

Richstokes. (s. f.). *GitHub - richstokes/cheekymonkey: 🐵 Literally a Chaos Monkey for your Kubernetes clusters*. GitHub. https://github.com/richstokes/cheekymonkey