

CLASE 47

En la computación en la nube, existen arquitecturas para la recuperación y detección dinámica de fallas como las que se exponen en el tema 10.7 del libro de texto y que se muestran en las siguientes figuras:

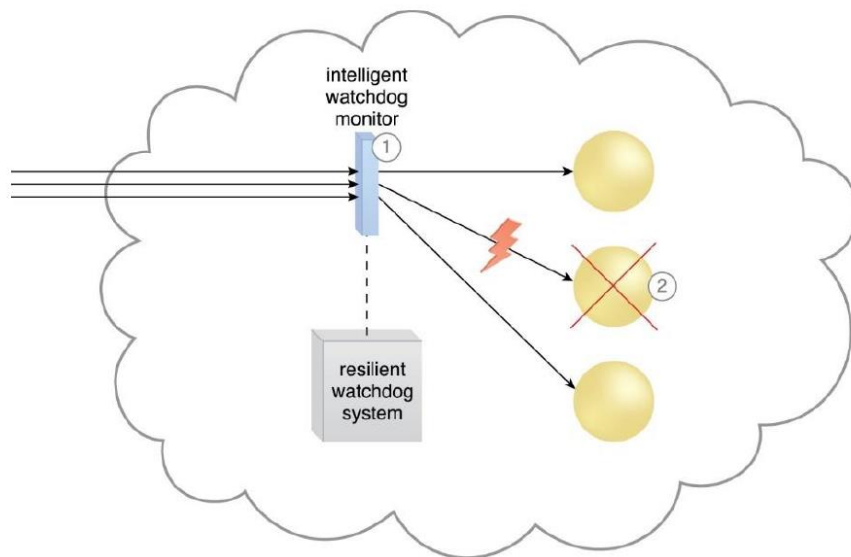


Figura 10.20 El monitor de watchdog inteligente realiza un seguimiento de las solicitudes de los consumidores de la nube (1) y detecta que un servicio de la nube ha fallado (2).

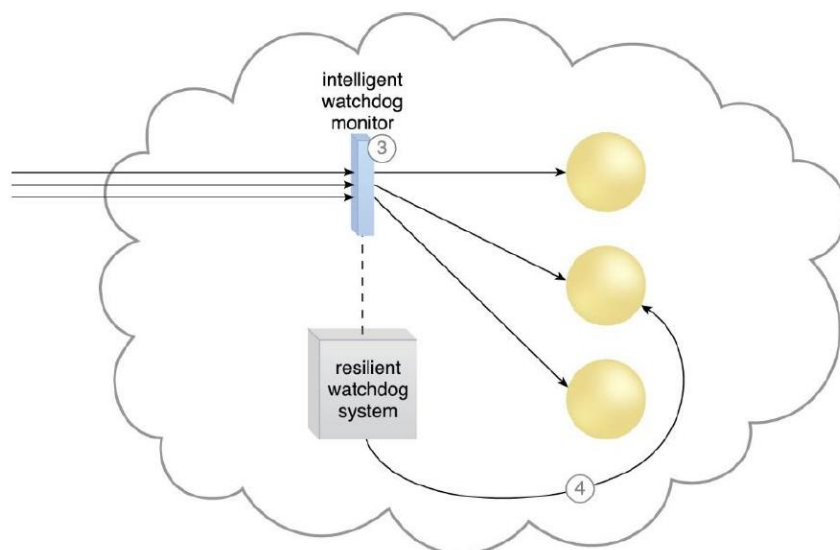
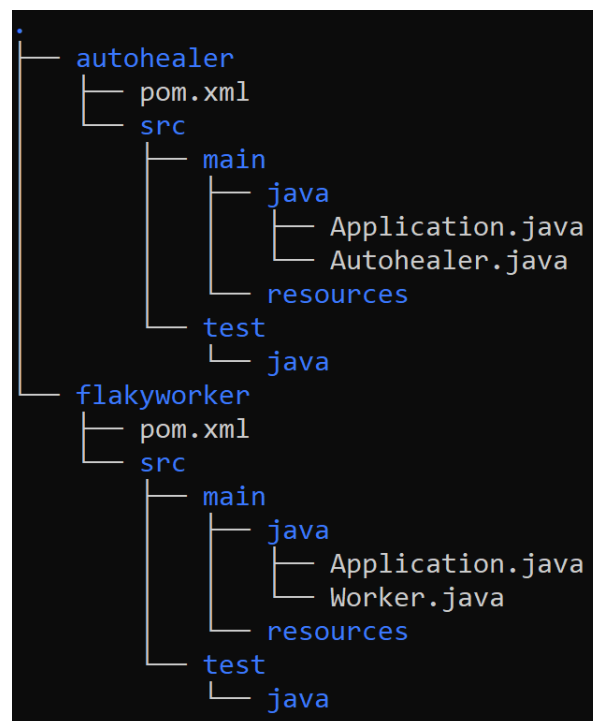


Figura 10.21 El monitor de watchdog inteligente notifica al sistema de watchdog (3), que restaura el servicio en la nube según políticas predefinidas. El servicio en la nube reanuda su funcionamiento en tiempo de ejecución (4).

En el código adjunto se implementa con ZooKeeper un auto reparador muy básico que monitorea un grupo de instancias de trabajadores. Se define al inicio el número de instancias que deben estar funcionando en todo momento y si alguna instancia falla el autoreparador (autohealer) crea una nueva instancia.

Las instancias de trabajadores simulan que están realizando un trabajo complejo que tiende a fallar debido a una situación no considerada por el programador.

Al descomprimir el archivo nos encontramos con dos proyectos como se muestra en la siguiente figura:



Cada uno de los proyectos se pueden compilar con el comando:

```
mvn clean install
```

Para su correcto funcionamiento se requiere primero levantar el servidor ZooKeeper en su versión 3.8.4.

Después de compilar ambos proyectos debemos ejecutar sólo el archivo jar creado en el proyecto autohealer y pasarle como parámetros el número de trabajadores que se desea, así como la ruta del archivo jar del trabajador (flakyworker). Por ejemplo, para tres trabajadores nos ubicamos dentro de la carpeta autohealer y ejecutamos:

```
java -jar ./target/autohealer-1.0-SNAPSHOT-jar-with-dependencies.jar 3
../flakyworker/target/flaky.worker-1.0-SNAPSHOT-jar-with-dependencies.jar
```

Ejercicio

Después de revisar y entender el funcionamiento de ambos proyectos, haga las modificaciones necesarias para ejecutar los trabajadores en una terminal o una computadora distinta a la terminal donde se ejecuta el autohealer.

Recomendaciones:

Crear un servicio http que se ejecutará en la otra terminal y cuando reciba un mensaje del autohealer creará un nuevo proceso (como lo hace el autohealer original) para ejecutar el código del nuevo trabajador (flakyworker).

El autohealer sólo se encargará de enviar un mensaje al servicio http cuando requiera crear un nuevo znode con ayuda de un cliente síncrono.

Considere cambiar el puerto si hay algún bloqueo en la comunicación. También es posible que el trabajador no imprima nada al ser creado por el servidor http.