

Diego Rodríguez Baquero  
201223538

## **Taller 8 – Estrategias y reglas monkeys**

**El chaos monkey tiene las siguientes estrategias:**

Shutdown instance: Mata instancias por completo.

Block all network traffic: Quita todos los grupos de seguridad y asigna uno sin acceso alguno

Burn CPU: Simula un carga excesiva de procesamiento por un vecino o un procesador dañado

Kill Processes: Mata todos los procesos de java/python, simulando aplicaciones rotas

Null route: Hace fallar todo el tráfico interno de EC2 all crear una ruta al vacío de la red interna

Fail EC2: Agrega entradas en el archivos *hosts* para hacer fallar la conexión con el API de EC2

Fail DNS: Bloquea el Puerto 53, hacienda fallar el DNS de la instancia

Fail DynamoDC: Agrega entradas en el archivos *hosts* para hacer fallar la conexión con DynamoDB

Fail S3: Agrega entradas en el archivos *hosts* para hacer fallar la conexión con S3

Network corruption: Corrompe una gran parte de los paquetes de red

Network latency: Simula una latencia de más o menos 1 segundo a todos los paquetes

Network loss: Simula una perdida de paquetes en red

Detach volumes: Forzadamente remueve discos EBS adjuntos a instancia, generando pérdida de datos

Burn IO: Corre procesos que intensos de operaciones de disco, generando latencia o simulando un disco con fallas

Fill disk: Escribe un archive muy grande a disco, llenándolo por completo

Burn money: No está documentado, pero incurre en no-pequeños gastos

## Reglas conformity monkey:

**SameZonesInElbAndAsg:** Verifica que los balanceadores de carga y el grupo de auto escalamiento estén en la misma zona de disponibilidad

**InstanceInSecurityGroup:** Verifica que todas las instancia estén en los grupos de seguridad requeridos, especificado en *requiredSecurityGroups*

**InstanceTooOld:** Verifica que las instancia no sean más viejas que *instanceAgeThreshold* días

**InstanceHasStatusUrl:** Verifica que todas las instancias tengan una URL de estado según Discovery/Eureka

**InstanceHasHealthCheckUrl:** Verifica que todas las instancias tengan una URL para revisar la salud de la instancia

**InstanceIsHealthyInEureka:** Verifica que las instancias estén saludables

## Reglas janitor monkey:

**orphanedInstanceRule:** Notifica al dueño de instancias que sean más viejas que *instanceAgeThreshold* días que se va a borrar la instancia. Si tiene dueño se elimina la instancia en *retentionDaysWithOwner* días. Si no tiene dueño, se borra en *retentionDaysWithoutOwner* días.

**untaggedRule:** Limpia recursos *resources* que no tengan los tags especificados en *requiredTags*. Se borra en *retentionDaysWithOwner* días si tiene dueño, en *retentionDaysWithoutOwner* días si no.

**oldDetachedVolumeRule:** Notifica de los volúmenes EBS que no pertenecen a ninguna instancia después de *detachDaysThreshold* días y los borra en *retentionDays* días si no hay respuesta.

**deleteOnTerminationRule:** Borra los recursos que debieron haber sido borrados por AWS al terminar un recursos tras *retentionDays* días de notificar.

**noGeneratedAMIRule:** Borra snapshots de más de *ageThreshold* días que no tengan AMI generado tras *retentionDays* días de notificar.

**oldEmptyASGRule:** Borra grupos de escalamiento sin instancias que tengan *launchConfigAgeThreshold* días de haber sido creadas tras *retentionDays* días de notificar.

**suspendedASGRule:** Borra grupos de escalamiento sin instancias y que hayan sido suspendidos del balanceador de carga asociado que tengan *suspensionAgeThreshold* días de haber sido suspendidas tras *retentionDays* días de notificar.

**oldUnusedLaunchConfigRule:** Borra configuraciones de lanzamiento sin uso que tengan *ageThreshold* días de creadas tras *retentionDays* días de notificar.

**unusedImageRule:** Borra imágenes que no se usan por instancias o configuraciones de lanzamiento y que no se hayan referenciado en *lastReferenceDaysThreshold* días, tras *retentionDays* días de notificar.