- Las diferentes estrategias de ataque del Chaos Monkey
    - Shutdown instance (Simius Mortus)
        - Apagar la instancia a través de EC2
    - Block all network traffic (Simius Quies)
        - Quita los permisos de los grupos de usuarios y los mueve a un grupo que no tiene acceso para acceder a los recursos.
    - Detach all EBS volumes (Simius Amputa)
        - Desconecta los EBS aunque la instancia siga corriendo. Puede causar perdida de datos.
    - Burn-CPU (Simius Cogitarius)
        - Aumenta el uso de la CPU a través de procesos intensivos para similar un uso excesivo de los recursos.
    - Burn-IO (Simius Occupatus)
        - Aumenta el uso del disco a través de procesos intensivos. La velocidad del disco se verá reducida significativamente.
    - Fill Disk (Simius Plenus)
        - Escribe un archivo grande en la memoria root llenándola prácticamente.
    - Kill Processes (Simius Delirius)
        - Mata procesos java o python que va encontrando para similar una instalación fallida o una instancia no estable.
    - Null-Route (Simius Desertus)
        - Este monkey null-route la red usada por EC2, queriendo decir que todo el tráfico dentro de esta red va a fallar.
    - Fail DNS (Simius Nonomenius)
        - Bloquear el puerto 53 TCP y UDP para similar una falla en los servicios DNS.
    - Fail EC2 API (Simius Nonneccius)
        - Similar al null-route, satura las conexiones con host bobos para que toda la comunicación EC2 falle.
    - Fail S3 API (Simius Amnesius)
        - Lo mismo que la anterior pero con la comunicacion dentro de S3.
    - Fail DynamoDB API (Simius Nodynamus)
        - Lo mismo que la anterior pero con la comunicacion dentro de DynnamoDB.
    - Network Corruption (Simius Politicus)
        - Toma una gran cantidad de los paquetes compartidos por la red y los corrompe para similar una degradación de la red.
    - Network Latency (Simius Tardus)
        - Este monkey usa la API de traffic shaping para añadir mayor tiempo de latencia en entrega de paquetes. También simula una degradación de la red
    - Network Loss (Simius Perditus)
        - Elimina cierta cantidad de los paquetes compartidos por la red. También simula una degradación de la red.
- Las reglas del Janitor Monkey

- Frequency: Cada cuanto va a realizar un análisis y limpieza de los recursos.
- Frequency unit: Que unidad se usara en el índice de frecuencia antes utilizado.
- Threads: cuantos procesos del janitor correrán en paralelo.
- Open hour: Desde que horas va a estar activado el janitor.
- Close hour: Hasta que horas va a estar activado el janitor.
- Time zone: En qué zona del mundo está para calcular el tiempo en el que las dos anteriores variables están definidas.
- Is monkey time: Activado o desactivado.
- Enabled: deja correr al monkey.
- Leashed: Puede o no el monkey apagar efectivamente las instancias de aws.
- Domain: en que dominio del simpleDB se van a guardar los recursos manejados por el monkey.
- Emails: Para determinar desde que dirección se van a enviar o recibir los distintos correos de notificaciones del monkey.
- Days before termination: Cuantos días deben pasar desde que el monkey notifica la terminación de una instancia hasta el momento en que efectivamente lo va a terminar.
- Enabled resources: Le dice al monkey a que recursos tiene acceso para hacer su magia.
- Orphaned Instance Rule Enabled: Le dice al monkey si puede limpiar instancias que no están como hijos de un ASG.
- Orphaned Instance Rule Instance Age Threshold: Determine the amount of days that an instances can run as orphan before being marked as target for clean up.
- orphanedInstanceRule.retentionDaysWithOwner: The number of business days the instance is kept after a notification is sent for the termination when the instance has an owner.
- orphanedInstanceRule.retentionDaysWithoutOwner: The number of business days the instance is kept after a notification is sent for the termination when the instance has no owner.
- orphanedInstanceRule.opsworks.parentage: If true, don't consider members of an OpsWorks stack as orphans
- untaggedRule.enabled: The following properties are used by the Janitor rule for cleaning up untagged resources, i.e. instances that are missing any required tags.
- untaggedRule.requiredTags: List of tags that are required for each resource
- untaggedRule.resources: List of resource types that require tags
- untaggedRule.retentionDaysWithOwner: The number of business days the resource is kept after a notification is sent for the deletion when the resource has an owner.
- untaggedRule.retentionDaysWithoutOwner: The number of business days the resource is kept after a notification is sent for the deletion when the resource has no owner.

-
- oldDetachedVolumeRule.enabled: The following properties are used by the Janitor rule for cleaning up volumes that have been detached from instances for certain days.
- oldDetachedVolumeRule.detachDaysThreshold: A volume is considered a cleanup candidate after being detached for the number of days specified in the property below.
- oldDetachedVolumeRule.retentionDays: The number of business days the volume is kept after a notification is sent for the termination.
- deleteOnTerminationRule.enabled: The following properties are used by the Janitor rule for cleaning up volumes that should have been deleted by AWS when the attached instance was terminated. This rule can be enabled only if Edda is enabled since Janitor Monkey needs to query the history of the attached instance.
- deleteOnTerminationRule.retentionDays: The number of business days the volume is kept after a notification is sent for the termination.
- noGeneratedAMIRule.enabled: The following properties are used by the Janitor rule for cleaning up snapshots that have no existing images generated from them and launched for certain days.
- noGeneratedAMIRule.ageThreshold: A snapshot without an image is considered a cleanup candidate after launching for the number of days specified in the property below.
- noGeneratedAMIRule.retentionDays: The number of business days the snapshot is kept after a notification is sent for the termination.
- oldEmptyASGRule.enabled: The following properties are used by the Janitor rule for cleaning up auto-scaling groups that have no active instances and the launch configuration is older than certain days.
- oldEmptyASGRule.launchConfigAgeThreshold: An an auto-scaling group without active instances is considered a cleanup candidate when its launch configuration is older than the number of days specified in the property below.
- simianarmy.janitor.rule.oldEmptyASGRule.retentionDays: The number of business days the auto-scaling group is kept after a notification is sent for the termination.
- suspendedASGRule.enabled: The following properties are used by the Janitor rule for cleaning up auto-scaling groups that have no active instances and have been suspended from the associated ELB traffic for certain days.
- suspendedASGRule.suspensionAgeThreshold: An auto-scaling group without active instances is considered a cleanup candidate when it has been suspended from the associated ELB traffic for the number of days specified in the property below.

    - suspendedASGRule.retentionDays: The number of business days the auto-scaling group is kept after a notification is sent for the termination.
    - oldUnusedLaunchConfigRule.enabled: The following properties are used by the Janitor rule for cleaning up launch configurations that are not used by any auto scaling group and are older than certain days.
    - oldUnusedLaunchConfigRule.ageThreshold: An unused launch configuration is considered a cleanup candidate when it is older than the number of days specified in the property below.
    - simianarmy.janitor.rule.oldUnusedLaunchConfigRule.retentionDays : The number of business days the launch configuration is kept after a notification is sent for the termination.
    - unusedImageRule.enabled: This rule is by default disabled, you need to have Edda running and enabled for using this rule.
    - unusedImageRule.lastReferenceDaysThreshold: An unused image is considered a cleanup candidate when it is not referenced for than the number of days specified in the property below.
    - unusedImageRule.retentionDays: The number of business days the image is kept after a notification is sent for the termination.
- Las reglas del Conformity Monkey
    - SameZonesInElbAndAsg.enabled: The following property is used to enable the conformity rule to check whether there is mismatch of availability zones between any auto scaling group and its ELBs in a cluster.
    - InstanceInSecurityGroup.enabled: The following property is used to enable the conformity rule to check whether all instances in the cluster are in required security groups.
    - InstanceInSecurityGroup.requiredSecurityGroups: The following property specifies the required security groups in the InstanceInSecurityGroup conformity rule.
    - InstanceTooOld.enabled: The following property is used to enable the conformity rule to check whether there is any instance that is older than certain days.
    - InstanceTooOld.instanceAgeThreshold: The following property specifies the number of days used in the InstanceInSecurityGroup, any instance that is old than this number of days is consider nonconforming.
    - InstanceHasStatusUrl.enabled: The following property is used to enable the conformity rule to check whether all instances in the cluster have a status url defined according to Discovery/Eureka.
    - InstanceHasHealthCheckUrl.enabled: The following property is used to enable the conformity rule to check whether all instances in the cluster have a health check url defined according to Discovery/Eureka.

- InstanceIsHealthyInEureka.enabled: The following property is used to enable the conformity rule to check whether there are unhealthy instances in the cluster according to Discovery/Eureka.