Taller 8 NetFlix Simian Army

David Ernesto Paniagua García

Las diferentes estrategias de ataque del Chaos Monkey

Shutdown instance (Simius Mortus)

Finaliza una instancia usando el API de EC2. Esta es la estrategia clásica de Chaos Monkey

Block all network traffic (Simius Quies)

Remueve los grupos de seguridad de la instancia y los mueve a otro grupo que no tenga acceso

Detach all EBS volumes *(Simius Amputa)*

Separa todos los volúmenes de EBS de la instancia, simulando una falla de EBS. Por lo tanto, la instancia se está ejecutando, pero la E / S del disco EBS fallará.

SSH monkeys

Este y los siguientes monkeys funcionan ejecutando scripts en la instancia. Para configurar, debe establecer la clave SSH que se utilizará para iniciar sesión en la instancia.

Nombre de usuario de SSH: simianarmy.chaos.ssh.user = root(El valor predeterminado es root) Ruta de la clave SSH: simianarmy.chaos.ssh.key = ~/monkeyboy/.ssh/id_rsa (ruta a su clave SSH)

Burn-CPU *(Simius Cogitarius)*

Este monkey ejecuta procesos intensivos de CPU, simulando un "vecino ruidoso" o una CPU defectuosa. La instancia tendrá efectivamente una CPU mucho más lenta.

Habilitar con: simianarmy.chaos.burncpu.enabled = true Requiere que se configure SSH.

Burn-IO *(Simius Occupatus)*

Este monkey ejecuta procesos intensivos en disco, simulando un "vecino ruidoso" o un disco defectuoso. La instancia tendrá efectivamente un disco mucho más lento.

Habilitar con: `simianarmy.chaos.burnio.enabled = true` Requiere que se configure SSH.

Fill Disk *(Simius Plenus)*

Este monkey escribe un archivo enorme en el dispositivo raíz, llenando el disco raíz EC2 (normalmente relativamente pequeño).

Habilitar con: `simianarmy.chaos.filldisk.enabled = true` Requiere que se configure SSH.

Kill Processes (Simius Delirius)

Este monkey mata cualquier programa java o python que encuentra cada segundo, simulando una aplicación defectuosa, una instalación dañada o una instancia defectuosa. La instancia está bien, pero la aplicación java / python que se ejecuta en ella fallará continuamente.

Habilitar con: `simianarmy.chaos.killprocesses.enabled = true` Requiere que se configure SSH.

Null-Route (Simius Desertus)

Este monkey enruta la red 10.0.0.0/8, que es utilizada por la red interna de EC2. Todo el tráfico de red EC2 <-> EC2 fallará.

Habilitar con: `simianarmy.chaos.nullroute.enabled = true` Requiere que se configure SSH.

Fail DNS (Simius Nonomenius)

Este monkey usa iptables para bloquear el puerto 53 para TCP y UDP; esos son los puertos de tráfico DNS. Esto simula una falla de sus servidores DNS.

Habilitar con: `simianarmy.chaos.faildns.enabled = true` Requiere que se configure SSH.

Fail EC2 API (Simius Noneccius)

Este monkey pone entradas de host ficticias en / etc / hosts, por lo que todas las comunicaciones de la API EC2 fallarán. Esto simula una falla del plano de control EC2. Por supuesto, si su aplicación no utiliza la API EC2 de la instancia, entonces no se verá afectada por completo.

Habilitar con: `simianarmy.chaos.failec2.enabled = true` Requiere que se configure SSH.

Fail S3 API (Simius Amnesius)

Este monkey pone entradas de host ficticias en / etc / hosts, por lo que toda la comunicación S3 fallará. Esto simula una falla de S3. Por supuesto, si su aplicación no usa S3, entonces no se verá afectada por completo.

Habilitar con: `simianarmy.chaos.fails3.enabled = true` Requiere que se configure SSH.

Fail DynamoDB API (Simius Nodynamus)

Este monkey pone entradas de host ficticias en / etc / hosts, por lo que todas las comunicaciones de DynamoDB fallarán. Esto simula una falla de DynamoDB. Al igual que con sus parientes de mono, esto solo afectará a las instancias que usen DynamoDB.

Habilitar con: `simianarmy.chaos.faildynamodb.enabled = true` Requiere que se configure SSH.

Network Corruption (Simius Politicus)

Este monkey usa la API de conformación de tráfico para coruplar una gran parte de los paquetes de red. Esto simula la degradación de la red EC2.

Habilitar con: `simianarmy.chaos.networkcorruption.enabled = true` Requiere que se configure SSH.

Network Latency (Simius Tardus)

Este monkey usa la API de conformación de tráfico para introducir latencia (1 segundo + - 50%) a todos los paquetes de red. Esto simula la degradación de la red EC2.

Habilitar con: `simianarmy.chaos.networklatency.enabled = true` Requiere que se configure SSH.

Network Loss (Simius Perditus)

Este monkey usa la API de conformación de tráfico para eliminar una fracción de todos los paquetes de red. Esto simula la degradación de la red EC2.

Habilitar con: `simianarmy.chaos.networkloss.enabled = true` Requiere que se configure SSH.

Las reglas del Janitor Monkey

Setting rule of cleaning up instances not in auto scaling group

The properties below are used to configure the rule used to clean up orphaned instances that are not in any auto scaling group.

simianarmy.janitor.rule.orphanedInstanceRule.enabled

This property specifies whether Janitor monkey will clean up orphaned instances. If you do not want to clean up orphaned instances, you can set the property to false to disable the rule. The default value is true.

```
simianarmy.janitor.rule.orphanedInstanceRule.enabled = true
```

simianarmy.janitor.rule.orphanedInstanceRule.instanceAgeThreshold

An orphaned instance is marked as cleanup candidate if it has launched for more than the number of days specified in this property. The default value is 2, which means orphaned instance is marked as cleanup candidate and scheduled to be terminated if it has launched for more than 2 days.

```
simianarmy.janitor.rule.orphanedInstanceRule.instanceAgeThreshold = 2
```

simianarmy.janitor.rule.orphanedInstanceRule.retentionDaysWithOwner

This property specifies the number of business days the instance is retained after a notification is sent about the termination when the instance has an owner. The default value is 3.

```
simianarmy.janitor.rule.orphanedInstanceRule.retentionDaysWithOwner = 3
```

simianarmy.janitor.rule.orphanedInstanceRule.retentionDaysWithoutOwner

This property specifies the number of business days the instance is retained after a notification is sent about the termination when the instance does not have an owner. The default value is 8.

```
simianarmy.janitor.rule.orphanedInstanceRule.retentionDaysWithoutOwner = 8
```

Setting rule of cleaning up detached EBS volumes

The properties below are used to configure the rule used for cleaning up volumes that have been detached from instances for certain days. Since AWS does not provide a way to track when an EBS volume is detached, we provide a companion monkey called **Volume Tagging Monkey** for tracking this information. You can modify **volumeTagging.properties** file to enable it. Details can be found at Configure Volume Tagging Monkey.

simianarmy.janitor.rule.oldDetachedVolumeRule.enabled

This property specifies whether Janitor monkey will clean up detached volumes. If you do not want to clean up detached volumes, you can set the property to false to disable the rule. The default value is true.

```
simianarmy.janitor.rule.oldDetachedVolumeRule.enabled = true
```

simianarmy.janitor.rule.oldDetachedVolumeRule.detachDaysThreshold

A volume is considered a cleanup candidate after being detached for the number of days specified in this property. The default value is 30.

```
simianarmy.janitor.rule.oldDetachedVolumeRule.detachDaysThreshold = 30
```

simianarmy.janitor.rule.oldDetachedVolumeRule.retentionDays

This property specifies the number of business days the volume is retained after a notification is sent about the termination. The default value is 7.

```
simianarmy.janitor.rule.oldDetachedVolumeRule.retentionDays = 7
```

Setting rule of cleaning up unreferenced snapshots

The properties below are used to configure the rule used for cleaning up snapshots that have no existing images generated from them and launched for certain days.

simianarmy.janitor.rule.noGeneratedAMIRule.enabled

This property specifies whether Janitor monkey will clean up unreferenced snapshots. You can set the property to false to disable the rule. The default value is true.

```
simianarmy.janitor.rule.noGeneratedAMIRule.enabled = true
```

simianarmy.janitor.rule.noGeneratedAMIRule.ageThreshold

A unreferenced snapshot is considered a cleanup candidate after launching for the number of days specified in this property. The default value is 30.

```
simianarmy.janitor.rule.noGeneratedAMIRule.ageThreshold = 30
```

simianarmy.janitor.rule.noGeneratedAMIRule.retentionDays

This property specifies the number of business days the snapshot is retained after a notification is sent about the termination. The default value is 7.

```
simianarmy.janitor.rule.noGeneratedAMIRule.retentionDays = 7
```

simianarmy.janitor.rule.noGeneratedAMIRule.ownerEmail

This property specifies an override owner email when cleaning EBS Snapshots with this rule. This will enable notification emails to be sent to a different email target rather than the resource owner. This could be useful for organizations that create a lot of snapshots as part of their build process and notifying owners is unnecessary.

The default value for this property is null meaning the resource owner will be notified and not the override value.

```
simianarmy.janitor.rule.noGeneratedAMIRule.ownerEmail = null
```

Setting rule of cleaning up empty auto scaling groups

The properties below are used to configure the rule used for cleaning up auto scaling groups that have no active instances and the launch configuration is more than certain days old.

simianarmy.janitor.Eureka.enabled

The property below specifies whether or not Eureka/Discovery is available for Janitor monkey to use. Discovery/Eureka is used in the rules for cleaning up auto scaling groups to determine if an auto scaling group has an 'active' instance, i.e. an instance that is registered and up in Discovery/Eureka. You should set this property to false if you do not have Discovery/Eureka set up in your environment. The default value of this property is false.

```
simianarmy.janitor.Eureka.enabled = false
```

simianarmy.janitor.rule.oldEmptyASGRule.enabled

This property specifies whether Janitor monkey will clean up empty auto scaling groups. You can set the property to false to disable the rule. The default value is true.

```
simianarmy.janitor.rule.oldEmptyASGRule.enabled = true
```

simianarmy.janitor.rule.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 this property. The default value is 50.

```
simianarmy.janitor.rule.oldEmptyASGRule.launchConfigAgeThreshold = 50
```

simianarmy.janitor.rule.oldEmptyASGRule.retentionDays

The number of business days an empty auto-scaling group is retained after a notification is sent for the termination. The default value is 10.

```
simianarmy.janitor.rule.oldEmptyASGRule.retentionDays = 10
```

Setting rule of cleaning up suspended auto scaling groups

The properties below are used to configure the rule used for cleaning up auto-scaling groups that have no active instances and have been suspended from the associated ELB traffic for certain days.

simianarmy.janitor.rule.suspendedASGRule.enabled

This property specifies whether Janitor monkey will clean up suspended auto scaling groups. You can set the property to false to disable the rule. The default value is true.

```
simianarmy.janitor.rule.suspendedASGRule.enabled = true
```

simianarmy.janitor.rule.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 this property, the default value is 2.

```
simianarmy.janitor.rule.suspendedASGRule.suspensionAgeThreshold = 2
```

simianarmy.janitor.rule.suspendedASGRule.retentionDays

This property specifies the number of business days the auto scaling group is retained after a notification is sent for the termination. The default value is 5.

```
simianarmy.janitor.rule.suspendedASGRule.retentionDays = 5
```

Setting rule of cleaning up launch configurations

The following properties are used by the Janitor rule for cleaning up launch configurations that are not used by any auto scaling group or instances and are older than certain days.

simianarmy.janitor.rule.oldUnusedLaunchConfigRule.enabled

This property specifies whether Janitor monkey will clean up unused launch configurations that are more than certain days old. You can set the property to false to disable the rule. The default value is true.

```
simianarmy.janitor.rule.oldUnusedLaunchConfigRule.enabled = true
```

simianarmy.janitor.rule.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. The default value is 4.

```
simianarmy.janitor.rule.oldUnusedLaunchConfigRule.ageThreshold = 4
```

simianarmy.janitor.rule.oldUnusedLaunchConfigRule.retentionDays

This property specifies the number of business days the launch configuration is kept after a notification is sent for the termination. The default value is 3.

```
simianarmy.janitor.rule.oldUnusedLaunchConfigRule.retentionDays = 3
```

Setting rule of cleaning up images

The following properties are used by the Janitor rule for cleaning up images.

simianarmy.janitor.image.crawler.lookBackDays

The property below specifies the number of days to look back in the history when crawling the last reference information of the images. By default we look back up to 60 days.

```
simianarmy.janitor.image.crawler.lookBackDays = 60
```

simianarmy.janitor.image.ownerId

The property below specifies the owner id that images have for being managed by Janitor Monkey. If no owner id is set, all images under the AWS account are returned. By default the line is commented and no owner id is set.

```
#simianarmy.janitor.image.ownerId = 1234567890
```

simianarmy.janitor.rule.unusedImageRule.enabled

This rule is used by the Janitor rule for cleaning up images that have not been used by any instances and launch configurations, and and not used to create other images, in the last certain days. This rule is by default disabled, you need to have Edda running and enabled for using this rule since the image's history is needed to determine the last time when the images was referenced.

```
simianarmy.janitor.rule.unusedImageRule.enabled = false
```

simianarmy.janitor.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. The default value is 45.

```
simianarmy.janitor.rule.unusedImageRule.lastReferenceDaysThreshold = 45
```

simianarmy.janitor.rule.unusedImageRule.retentionDays

The property below specifies the number of business days the image is kept after a notification is sent for the termination. The default value is 3.

```
simianarmy.janitor.rule.unusedImageRule.retentionDays = 3
```

Las reglas del Conformity Monkey

Conformity Properties

With these properties you can control how conformity monkey operates.

simianarmy.conformity.enabled

This property allows the Conformity Monkey to run. The default is "true". Note that an enabled Conformity Monkey does not mean it saves the conformity check results to DB or send notifications for unconforming clusters. It must also have the leashed property set to "false".

```
simianarmy.conformity.enabled = true
```

Customizing Conformity Monkey's schedule

By default Conformity Monkey wakes up and performs conformity checks, and sends notifications every hour. The properties below override the values set in **simianarmy.properties**.

```
simianarmy.scheduler.frequency = 1
simianarmy.scheduler.frequencyUnit = HOURS
simianarmy.scheduler.threads = 1
simianarmy.calendar.openHour = 0
simianarmy.calendar.closeHour = 24
simianarmy.calendar.timezone = America/Los_Angeles
```

####simianarmy.calendar.isMonkeyTime To ease debug/dryrun, you can set this property to true so the monkey will always be able to run. This property also overrides the value in **simianarmy.properties**. The default value is false.

```
simianarmy.calendar.isMonkeyTime = false
```

The properties below allow you to customize whether conformity monkey sends notifications for unconforming clusters every hour or only within a specific time window. Conformity monkey sends notifications to the owner of unconforming clusters between the open hour and close hour only. In other hours, only summary email is sent. The default setting is to always send email notifications after each run.

```
simianarmy.conformity.notification.openHour = 0
simianarmy.conformity.notification.closeHour = 24
```

simianarmy.conformity.leashed

This is effectively a "dryrun" option. If leashed is "true" then it will performs conformity checks for all clusters, but it will not be able to save data to DB or send notifications. The default value is "true".

```
simianarmy.conformity.leashed = true
```

simianarmy.conformity.resources.sdb.domain

This property specifies the SimpleDB domain for storing the data managed by the Conformity Monkey. The default is SIMIAN_ARMY.

```
simianarmy.conformity.sdb.domain = SIMIAN_ARMY
```

simianarmy.conformity.notification.sourceEmail

This property specifies the source email address that is used to send notifications about unconforming clusters or the summery email. Its value needs to be a valid email address.

```
simianarmy.conformity.notification.sourceEmail = foo@bar.com
```

simianarmy.conformity.summaryEmail.to

This property specifies the email address to receive the summary email of Conformity Monkey after each run. The property value needs to be a valid email address.

```
simianarmy.conformity.summaryEmail.to = foo@bar.com
```

simianarmy.conformity.notification.defaultEmail

This property specifies the email address to receive notification about unconforming clusters when Conformity Monkey is not able to find the owner email. You can set this property to your team's email list so that your team members are able to receive Conformity Monkey notifications.

```
simianarmy.conformity.notification.defaultEmail = foo@bar.com
```

simianarmy.conformity.Eureka.enabled

This property is for enabling the conformity rules that need to access Eureka for Conformity Monkey. By default Eureka is not enabled. You can only enable it if you have set up Eureka running otherwise you will see exceptions when Conformity Monkey tries to initialize Eureka client.

```
simianarmy.conformity.Eureka.enabled = false
```

simianarmy.conformity.rule.SameZonesInElbAndAsg.enabled

This property is used to enable the conformity rule to check whether there is mismatch of availability zones between any auto scaling group and its elastic load balancers in a cluster.

```
simianarmy.conformity.rule.SameZonesInElbAndAsg.enabled = true
```

simianarmy.conformity.rule.InstanceInSecurityGroup.enabled

This property is used to enable the conformity rule to check whether all instances in a cluster are in required security groups.

```
simianarmy.conformity.rule.InstanceInSecurityGroup.enabled = true
```

simianarmy.conformity.rule.InstanceInSecurityGroup.requiredSecurityGroups

This property specifies the required security groups in the InstanceInSecurityGroup conformity rule.

```
simianarmy.conformity.rule.InstanceInSecurityGroup.requiredSecurityGroups = foo, bar
```

simianarmy.conformity.rule.InstanceTooOld.enabled

This property is used to enable the conformity rule to check whether there is any instance that is older than certain days.

```
simianarmy.conformity.rule.InstanceTooOld.enabled = true
```

simianarmy.conformity.rule.InstanceTooOld.instanceAgeThreshold

This property specifies the number of days used in the InstanceInSecurityGroup, any instance that is old than this number of days is consider nonconforming.

```
simianarmy.conformity.rule.InstanceTooOld.instanceAgeThreshold = 180
```

simianarmy.conformity.rule.InstanceInVPC.enabled

This property is used to enable the conformity rule to check whether your instances are in an Amazon Virtual Private Cloud (VPC).

```
simianarmy.conformity.rule.InstanceInVPC.enabled = true
```

simianarmy.conformity.rule.InstanceHasStatusUrl.enabled

This property is used to enable the conformity rule to check whether all instances in the cluster have a status url defined in Discovery/Eureka. The rule is added to Conformity Monkey only when Eureka is enabled also.

```
simianarmy.conformity.rule.InstanceHasStatusUrl.enabled = true
```

simianarmy.conformity.rule.InstanceHasHealthCheckUrl.enabled

This property is used to enable the conformity rule to check whether all instances in the cluster have a health check url defined in Discovery/Eureka. The rule is added to Conformity Monkey only when Eureka is enabled also.

```
simianarmy.conformity.rule.InstanceHasHealthCheckUrl.enabled = true
```

simianarmy.conformity.cluster.name.ownerEmail

You can override a cluster's owner email by providing a property here. For example, the line below overrides the owner email of cluster foo to foo@bar.com

```
simianarmy.conformity.cluster.foo.ownerEmail = foo@bar.com
```

simianarmy.conformity.cluster.name.excludedRules

You can exclude specific conformity rules for a cluster using this property. For example, the line below excludes the conformity rule rule1 and rule2 on cluster foo.

```
simianarmy.conformity.cluster.foo.excludedRules = rule1,rule2
```

simianarmy.conformity.cluster.name.optedOut

You can opt out a cluster completely from Conformity Monkey by using this property. After a cluster is opted out, no notification is sent for it no matter it is conforming or not. For example, the line below opts out the cluster foo.

```
simianarmy.conformity.cluster.foo.optedOut = true
```

simianarmy.conformity.cluster.SoloInstances.optedOut

The 'SoloInstances' cluster is a specific cluster containing all the instances that you own that are not part of an auto-scaling group. Currently the only conformity check applied to these instances is the 'InstanceInVPC' rule, however others can be modified to work against this cluster. If you wish to opt out add this property.

```
simianarmy.conformity.cluster.SoloInstances.optedOut = true
```

Bibliografía

https://github.com/netflix/simianarmy/wiki/The-Chaos-Monkey-Army

https://github.com/netflix/simianarmy/wiki/janitor-settings

https://github.com/netflix/simianarmy/wiki/conformity-settings