categories: date: 2018-08-21 description: published: false sources: title: "Outline: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training"

Outline: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training

Purpose

Content is aimed at subtly nudging readers away from initial inquiries of "what is Chaos Monkey and how might it benefit my organization?" toward the more robust, nuanced, and educated view that "Chaos Monkey is useful for what it does, but it may not be the proper solution for all my organization's needs. Other solutions (such as Gremlin) are often better suited for our purposes." *A smart engineer will infer that the answer to whether to use Chaos Monkey or another solution is, "Well, it depends." This guide enables and encourages intelligent readers to consider such possibilities.*

Site Map

The following is the initial rough site map for the project, illustrating the parent/child relationship between the `pillar page`, `hub pages`, `category pages`, and `sections`. The `pillar page` is the base or "root" page of the entire Chaos Monkey web content structure. `Hub pages` contain larger collections of content about each relevant topic, and will include the majority of tangible, actionable content. `Category sections` are sub-sections contained within their parent `hub page`, with each detailing the specifics of the categorized topic, including a multitude of `sections` within each `category`. Each `section` is a small written piece about the given topic, which is contained within the parent `category`. In cases of a combined `section + page`, the `section` blurb will be contained in the parent `page`, while additional article content will be found in a unique `page` URL relevant to that `section`.

Unless otherwise discussed or later modified, generally each `page` within this site map will be a full article with a unique URL, while each `section` is a small written piece of content to be inserted into the larger, parent `page`:

- Pillar Page: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training (`https://www.gremlin.com/chaos-monkey`)
  - Hub Page: The Origin of Chaos Monkey - Why Netflix Needed to Create Failure (`https://www.gremlin.com/chaos-monkey/origin-netflix`)
  - Hub Page: Chaos Monkey Tutorial - A Step-by-Step Guide to Creating Failure on AWS (`https://www.gremlin.com/chaos-monkey/developer-tutorial`)
  - Hub Page: Taking Chaos Monkey to the Next Level - Advanced Developer Guide (`https://www.gremlin.com/chaos-monkey/advanced-tips`)
  - Hub Page: The Simian Army - Overview and Resources (`https://www.gremlin.com/chaos-monkey/simian-army`)
  - Hub Page: Chaos Monkey Resources, Guides, and Downloads for Engineers (`https://www.gremlin.com/chaos-monkey/downloads-resources`)

  - Hub Page: Chaos Monkey Alternatives - Tools for Creating Chaos Outside of AWS (`https://www.gremlin.com/chaos-monkey/alternatives`)
    - Category Section: Apache
      - Section: Hadoop
      - Section: Kafka
      - Section: Spark
    - Category Section: Containers
      - Section + Page: Docker (`https://www.gremlin.com/chaos-monkey/alternatives/docker`)
      - Section + Page: Gremlin (`https://www.gremlin.com/chaos-monkey/alternatives/gremlin`)
    - Category Section: Erlang VM
      - Section: Elixir
    - Category Section: Infrastructure
      - Section + Page: Azure (`https://www.gremlin.com/chaos-monkey/alternatives/azure`)
      - Section + Page: Google Cloud Platform (`https://www.gremlin.com/chaos-monkey/alternatives/google-cloud-platform`)
      - Section + Page: Kubernetes (`https://www.gremlin.com/chaos-monkey/alternatives/kubernetes`)
      - Section: On Premise
      - Section + Page: OpenShift (`https://www.gremlin.com/chaos-monkey/alternatives/openshift`)
      - Section: OpenStack
      - Section + Page: Private Cloud (`https://www.gremlin.com/chaos-monkey/alternatives/private-cloud`)
      - Section + Page: VMWare (`https://www.gremlin.com/chaos-monkey/alternatives/vmware`)
    - Category Section: Java
      - Section: Maven
      - Section + Page: Spring Boot (`https://www.gremlin.com/chaos-monkey/alternatives/spring-boot`)
    - Category Section: OS
      - Section: Linux
      - Section: Windows

Each `hub page` (i.e. `/chaos-monkey/[hub-page]`) should be quite in-depth in terms of both content length and topic coverage (per `Topics` column in spreadsheet).

Alternative `category pages` with separate URLs (i.e. `https://www.gremlin.com/chaos-monkey/alternatives/[technology]`) should also include full articles, but will likely be relatively narrower in scope compared to `hub pages`.

Outline

The following outlines the proposed content within each `page` and/or `section` of the proposed `site map`, and will be loosely used and referenced during the research and creation of the finalized content.

Pillar Page: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training

- URL: `https://www.gremlin.com/chaos-monkey/`
- Parent: None
- Content:
  - `Intro`
    - Who: Explain that the guide is meant for all audiences, including engineers, managers, executives, hobbyists, students, and so forth; anyone who is looking to learn more about what Chaos Monkey can (and cannot) do.
    - What: Summarize what the guide will cover, from `hub page` sections such as step-by-step tutorials, advanced engineering tips, simian army details, extensive tool recommendations, and an abundance of additional resources, all curated by Gremlin.
    - Why: Discuss the purpose of the guide to provide resources, critical information, tools, and insight from leaders in the field, so engineers and other readers can make educated decisions about their own Chaos Engineering path.
    - When/Where: Briefly explore when and where this guide can be used -- along with Chaos Monkey and Chaos Engineering in general -- including both professional and private scenarios. Mention of downloadable `PDF` guide for offline viewing as well.
  - `Pros` and `Cons` list for Chaos Monkey. Articulate the subtle limitations of Chaos Monkey, while educating readers about alternatives.
    - Describe Chaos Monkey's usefulness for certain types of failures (e.g. randomized AWS instance unavailability), but express the limitations this brings.
  - `Hub Page Section Blurbs`: Include `section` blurbs introducing each `hub page` topic.
    - `Section: The Origin of Chaos Monkey - Why Netflix Needed to Create Failure`
    - `Section: Chaos Monkey Tutorial - A Step-by-Step Guide to Creating Failure on AWS`
    - `Section: Taking Chaos Monkey to the Next Level - Advanced Developer Guide`
    - `Section: The Simian Army - Overview and Resources`
    - `Section: Chaos Monkey Resources, Guides, and Downloads for Engineers`
    - `Section: Chaos Monkey Alternatives - Tools for Creating Chaos Outside of AWS`

Hub Page: The Origin of Chaos Monkey - Why Netflix Needed to Create Failure

- URL: `https://www.gremlin.com/chaos-monkey/origin-netflix`
- Parent: `Pillar Page: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training`
- Content:
  - `Intro`
    - Why: Answer the general question proposed in title as to why Netflix needed to create Chaos Monkey in the first place. Detail the overall creation story.
    - What: Establish the current (and historic) role of Chaos Monkey within the Netflix architecture and engineering practices.
    - When/Where: Explore example scenarios where Chaos Monkey has been used and is most appropriate.

    - Who: Discuss other organizations using Chaos Monkey (Amazon, Spinnaker, etc). Expound on the internal culture that Chaos Monkey helped to promote at both Netflix, Amazon, and elsewhere.
  - The Simian Army
    - Briefly explore Simian Army, as introduction and lead into The Simian Army - Overview and Resources hub page.
  - Chaos Monkey Today
    - Examine modern and future Chaos Monkey iterations (2.0+).
    - Explain changes and upgrade paths from 1.0 to 2.0.
    - Explore Netflix Failure Injection Testing (FIT)
    - Discuss Netflix's (and others') use of Spinnaker and the potential issues and limitations such tools may present (investment in setup and propagation, maintenance, forced continuous delivery solution, etc).

Hub Page: Chaos Monkey Tutorial - A Step-by-Step Guide to Creating Failure on AWS

- URL: https://www.gremlin.com/chaos-monkey/developer-tutorial
- Parent: Pillar Page: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training
- Content:
  - Intro
    - What: Describe the purpose of the tutorial, which is to give not only a step-by-step guide on setting up and using Chaos Monkey with AWS, but we'll also go beyond initial setup to explore specific *scenarios* in which Chaos Monkey may (or may not) be relevant.
    - Who: Define intended audiences (primarily technically-minded professionals or hobbyists, engineers, developers, etc).
    - When/Why: The entire page will expound on the *when* and *why*, detailing scenarios (as listed below) in which using Chaos Monkey is appropriate, and providing the *how to* technical steps to make it work.
  - Basic Usage Guide
    - Provide full walk through detailing propagation, configuration, setup, and execution of Chaos Monkey within an AWS instance.
  - Monitoring Health and Stability
  - Detecting Service Latency
  - Discovering Security Vulnerabilities
  - Examining Exception Handling
  - On Demand Chaos
  - Building Your Own Customized Monkey
  - Additional Resources: Section to include an abundance of additional links and resources, with careful consideration for *types* of curated content. Resources should include written tutorials, books, research papers, talks/conferences, podcasts, videos, and so forth.

Hub Page: Taking Chaos Monkey to the Next Level - Advanced Developer Guide

- URL: https://www.gremlin.com/chaos-monkey/advanced-tips
- Parent: Pillar Page: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training
- Content:
  - Per **Austin**, content to be determined after direction from **Kolton**.

Hub Page: The Simian Army - Overview and Resources

- URL: `https://www.gremlin.com/chaos-monkey/simian-army`
- Parent: `Pillar Page: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training`
- Content:
  - `Intro`
    - What: Detail what the Simian Army is and how it relates to the more well known topic of Chaos Monkey.
    - Who: Define the intended audience of engineers, developers, and so forth looking to educate themselves.
    - Why: Describe the purpose of the page, which is as an extensive collection of resources related to the broader Simian Army of tools and concepts.
    - When/Where: Given the extent of the resources and information provided, this page can be used within all manner of scenarios and for virtually any organization. Include CTA to full downloadable `PDF` guide for offline viewing.
  - `Simian Army Variants`: Provide a brief overview of each individual `variant` within the Simian Army, as listed below.
    - Chaos Gorilla
    - Chaos Kong
    - Latency Monkey
    - Doctor Monkey
    - Janitor Monkey
    - Conformity Monkey
    - Security Monkey
    - 10-18 Monkey
  - `Simian Army Strategies`: List and examine each of the dozen or so toggleable `strategies` defined here.

Hub Page: Chaos Monkey Resources, Guides, and Downloads for Engineers

- URL: `https://www.gremlin.com/chaos-monkey/downloads-resources`
- Parent: `Pillar Page: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training`
- Layout: Model based on this resource collection.
- Content:
  - `Intro`
    - What: The goal of this `page` is to provide the full list of resources, tools, and other links discovered during research on this project, all of which should be valid and valuable to readers of all kinds.
    - Who: All audiences are intended and welcome, including both technical and non-technical readers.
    - Why: To provide useful resources for anyone trying to get started with Chaos Monkey and Chaos Engineering.
  - `Resources`: Unless otherwise specified, every helpful and properly curated resource should be categorized and included in this list. Generally, links should not be given any contextual explanation. Instead, each `category` can include a brief explanation and overview to assist the reader in finding the appropriate resources via a quick scan or table of contents link. *Note: The*

following `categories` are not finalized and can easily be modified as necessary. Additionally, each `category` can be split by resource `type`, such as `text`, `audio`, `video` sub-sections.

    - Blogs
        - `Audio`
        - `Text`
        - `Video`
    - Books
    - Community
    - Conferences & Meetups
    - Discussion
    - Research Papers
    - Social Media
    - Tools
    - Tutorials
  - `Additional Chaos Monkey Resources`: In addition to outside resources, *all* unique pages/URLs found within this guide should be included in this section. It can later be determined how best to format this, but one possibility is a separate section containing a `site map` link collection.

Hub Page: Chaos Monkey Alternatives - Tools for Creating Chaos Outside of AWS

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Pillar Page: Chaos Monkey Guide for Engineers - Tips, Tutorials, and Training`
- Layout: Model based on this landing page list. A top-level `table of contents` should be anchored to `categories` and `sections` within the content.
- Content:
  - `Intro`
    - What: Describe the purpose of this page as a one-stop resource for a multitude of alternatives to Chaos Monkey, curated and sorted based on the primary technology on which they are based.
    - Who: Explain that the intended audience is engineers looking for alternative solutions to Chaos Monkey that can be used outside of AWS. Whether the reader knows the specific technology they wish to use, or not, these resources will help.
    - Why: To provide a list of Chaos Monkey alternatives for readers looking to learn, or who may be disappointed in Chaos Monkey's limitations.
  - `Categories`: As listed below, each relevant category (as defined in this spreadsheet) will be given its own `section` with a brief description and overview. Within each `category section` is all relevant sub-technology `sections`, which will each be detailed as necessary. When relevant, a given `section` should contain a link to the explicit `/alternatives/<technology>` page, which will provide additional details about the actual implementation of a Chaos Monkey alternatives within said technology.
  - `Additional Chaos Monkey Resources`: In addition to outside resources, *all* unique pages/URLs found within this guide should be included in this section. It can later be determined how best to format this, but one possibility is a separate section containing a `site map` link collection.

Category Section: Apache

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Pillar Page: Chaos Monkey - The Complete Guide`
- Content:
  - `Overview`: Each technology `section` will contain a brief overview, followed by the list of relevant examples and case studies where Chaos Engineering practices have been implemented within said technology.
  - `Apache Mesos & Marathon`: Discuss using `Apache Mesos` to abstract CPU, memory, IO, and other compute operations away from machines, while also using `Marathon` for container orchestration, which allow for simple REST API calls to initiate Chaos Experiments in the system.

Section: Hadoop

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Category Section: Apache`
- Content:
  - `Overview`: *See above.*
  - `Testing Hadoop Cluster Stability`: Detail techniques in which nodes within a Hadoop cluster may be taken down via Chaos experiments, with the possibly of including `High Availability` components.

Section: Kafka

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Category Section: Apache`
- Content:
  - `Overview`: *See above.*
  - `How ZipRecruiter Uses Chaos Engineering to Prevent Production Incidents`: Discuss how ZipRecruiter uses Chaos Engineering with Kafka to improve stability and overcome failures, including cluster updating, rolling upgrades vs blue/green vs active/passive approaches, and more.
  - `Testing Kafka Message Deletion by Topic`: Outline how Chaos Engineering can be used to test the deletion of `messages` within specific `topics` inside a Kafka distribution.

Section: Spark

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Category Section: Apache`
- Content:
  - `Overview`: *See above.*
  - `How Netflix Uses Chaos Monkey to Attack Spark Streaming`: Post details how Netflix's use of Apache Spark Streaming in the AWS cloud is tested through various Chaos Experiments.

Category Section: Containers

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Hub Page: Chaos Monkey Alternatives - Tools for Creating Chaos Outside of AWS`
- Content:

  - `Overview`: *See above.*

Section + Page: Docker

- URL: `https://www.gremlin.com/chaos-monkey/alternatives/docker`
- Parent: `Category Section: Containers`
- Content:
  - `Overview`: *See above.*
  - `docker-chaos-monkey`: Overview of the `docker-chaos-monkey` tool for implementing Chaos Monkey systems within a `Docker Swarm`.
  - `docker-simianarmy`: A simple `Docker` image of the Simian Army toolset.
  - `pumba`: A CLI chaos testing and network emulation tool for `Docker` that uses Linux kernel traffic controls to handle underlying network commands.

Section + Page: Gremlin

Note: Gabe added this section temporarily, until approval or request for removal. Uncertain if Gremlin wishes to include an `alternative` solution within the list directed at their own services. If section should be kept, Gabe will need Gremlin demo account to play around with and learn from, to assist with knowledge during writing of this section.

- URL: `https://www.gremlin.com/chaos-monkey/alternatives/gremlin`
- Parent: `Category Section: Containers`
- Content:
  - `Overview`: *See above.*
  - `Using Gremlin ...`: *See `Note` above.*

Category Section: Erlang VM

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Hub Page: Chaos Monkey Alternatives - Tools for Creating Chaos Outside of AWS`
- Content:
  - `Overview`: *See above.*
  - `Attacking ErlangVM Running 1000 Parallel Actors`: Talk and demo describing how Chaos Monkeys can be used within an ErlangVM instance to produce unpredictable and partially critical situations within the system.

Section: Elixir

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Category Section: Erlang VM`
- Content:
  - `Overview`: *See above.*
  - `chaosmonkey-elixir`: A simple `Elixir` Chaos Monkey implementation designed to randomly kill `Elixir` processes.
  - `chaos-spawn`: Similar to `chaosmonkey-elixir`, `chaos-spawn` periodically terminates random `Elixir` processes. Allows for some additional customization to determine which processes are

considered valid targets.

Category Section: Infrastructure

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Hub Page: Chaos Monkey Alternatives - Tools for Creating Chaos Outside of AWS`
- Content:
  - `Overview`: *See above.*

Section + Page: Azure

- URL: `https://www.gremlin.com/chaos-monkey/alternatives/azure`
- Parent: `Category Section: Infrastructure`
- Content:
  - `Overview`: *See above.*
  - `Azure Search`: Detail how Microsoft utilizes multiple `Chaos levels` (low, medium, high) to categorize and inform severity and actionability of failures within the Azure Search systems.
  - `WazMonkey`: Provide details on `WazMonkey`, an alternative to Chaos Monkey specifically designed for testing resiliency within Azure cloud services.
  - `Fault Analysis Service`: Provide an overview of Azure's `Fault Analysis Service`, which is designed for testing services built atop `Azure Service Fabric`.

Section + Page: Google Cloud Platform

- URL: `https://www.gremlin.com/chaos-monkey/alternatives/google-cloud-platform`
- Parent: `Category Section: Infrastructure`
- Content:
  - `Overview`: *See above.*
  - `Engineering Chaos Experiments with TensorFlow`: Overview of using Gremlin and distributed TensorFlow cluster on `Google Cloud Platform` to perform Chaos experiments.
  - `Deploying Chaos Monkey on Google Compute Engine`: Tutorial for integrating Chaos Monkey within `Google Compute Engine` using `Spinnaker` and `Golang`.

Section + Page: Kubernetes

- URL: `https://www.gremlin.com/chaos-monkey/alternatives/kubernetes`
- Parent: `Category Section: Infrastructure`
- Content:
  - `Overview`: *See above.*
  - `kube-monkey`: Detail the `kube-monkey` tool, which is an implementation of Chaos Monkey for Kubernetes clusters.
  - `Exploring Attack Resiliency via Node Maintenance`: Overview of how to use the open source `Chaos Toolkit` to inject Chaos experiments into Kubernetes clusters.
  - `Database Termination Experiment with Kubernetes and PostgreSQL`: Tutorial for how to test the loss of the master database and its impact on the overall application.
  - `pod-reaper`: A rule-based pod killing container for Kubernetes.

  - `powerfulseal`: Adds Chaos experiments to Kubernetes via targeted pod killing and starting/stopping VMs.

Section: On Premise

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Category Section: Infrastructure`
- Content:
  - `Overview`: *See above.*
  - `powerfulseal`: Adds Chaos experiments to local machines via targeted pod killing and starting/stopping processes.

Section + Page: OpenShift

- URL: `https://www.gremlin.com/chaos-monkey/alternatives/openshift`
- Parent: `Category Section: Infrastructure`
- Content:
  - `Overview`: *See above.*
  - `monkey-ops`: Describe how the `monkey-ops` service can be used to perform Chaos Engineering tasks inside an `OpenShift` container application.

Section: OpenStack

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Category Section: Infrastructure`
- Content:
  - `Overview`: *See above.*
  - `powerfulseal`: Adds Chaos experiments to OpenStack systems via targeted pod killing and starting/stopping VMs.

Section + Page: Private Cloud

- URL: `https://www.gremlin.com/chaos-monkey/alternatives/private-cloud`
- Parent: `Category Section: Infrastructure`
- Content:
  - `Overview`: *See above.*
  - `GomJabbar`: Detail the `GomJabbar` service, aimed at providing Chaos Monkey-like services and tools within a private cloud architecture.

Section + Page: VMWare

- URL: `https://www.gremlin.com/chaos-monkey/alternatives/vmware`
- Parent: `Category Section: Infrastructure`
- Content:
  - `Overview`: *See above.*

- `chaos-lemur`: A local application for random virtual machine destruction in BOSH-managed environments including `vSphere`/`VMWare`, by using included infrastructure APIs.

Category Section: Java

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Hub Page: Chaos Monkey Alternatives - Tools for Creating Chaos Outside of AWS`
- Content:
  - `Overview`: *See above.*
  - `byte-monkey`: A small library for performing `fault`, `latency`, `nullify`, and `short-circuit` Chaos experiments within JVM applications.

Section: Maven

*Note: Gabe is unclear about the implementation potential of Chaos Engineering practices with Maven. Need clarification from **Kolton** or others whether this is a valid section to expound upon.*

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Category Section: Java`
- Content:
  - `Overview`: *See above.*

Section + Page: Spring Boot

- URL: `https://www.gremlin.com/chaos-monkey/alternatives/spring-boot`
- Parent: `Category Section: Java`
- Content:
  - `Overview`: *See above.*
  - `Spring Boot Chaos Monkey`: A small library which implements a Chaos Monkey for testing and attacking a `Spring Boot` application.

Category Section: OS

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Hub Page: Chaos Monkey Alternatives - Tools for Creating Chaos Outside of AWS`
- Content:
  - `Overview`: *See above.*

Section: Linux

Note: Gabe is unclear how to approach adding Chaos Monkey alternative technologies within the broad topics of "Linux" and/or "Windows" OSes. Should previously-mentioned tools and techniques be mentioned here where applicable, or do we need unique suggestions not found elsewhere in the `alternatives` pages? Will seek advisement from Austin/Kolton.

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Category Section: OS`
- Content:

  - `Overview`: *See above.*

Section: Windows

Note: See above `Section: Linux` note.

- URL: `https://www.gremlin.com/chaos-monkey/alternatives`
- Parent: `Category Section: OS`
- Content:
  - `Overview`: *See above.*

Sources

Following is the current list of sources used during creation of this content. Any/all of these resources may be included in the full `Resources` sections in the final written product, but formatting will be modified and additional details will be provided wherever relevant.

- Chaos Engineering
  - General
    - https://github.com/dastergon/awesome-chaos-engineering
    - https://people.ucsc.edu/~palvaro/fit-ldfi.pdf
    - https://www.gremlin.com/community/tutorials/chaos-engineering-the-history-principles-and-practice/
    - https://coggle.it/diagram/WiKceGDAwgABrmyv/t/chaos-engineering-companies%2C-people%2C-tools-practices/0a2d4968c94723e48e1256e67df51d0f4217027143924b23517832f53c536e62
    - https://slofile.com/slack/chaosengineering
    - https://www.gremlin.com/community/tutorials/4-chaos-experiments-to-start-with/
    - https://chaostoolkit.org/
  - Apache
    - http://www.datio.com/iaas/chaos-engineering-and-mesos/
    - http://mesos.apache.org/
    - https://mesosphere.github.io/marathon/
  - Azure
    - General
      - https://azure.microsoft.com/en-gb/blog/inside-azure-search-chaos-engineering/
      - https://azure.microsoft.com/en-us/services/search/
      - https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-testability-scenarios
      - https://www.infoq.com/news/2012/09/azure-chaos-monkey
      - https://github.com/smarx/WazMonkey
    - Azure Service Fabric
      - https://azure.microsoft.com/en-us/services/service-fabric/
      - https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview
      - https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-testability-overview

      - https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-controlled-chaos
      - https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-testability-scenarios
  - Docker
    - https://codefresh.io/docker-tutorial/chaos_testing_docker/
    - https://github.com/alexei-led/pumba
    - https://leanpub.com/12fa-docker-golang
    - https://github.com/mlafeldt/docker-simianarmy
    - https://github.com/titpetric/docker-chaos-monkey
  - Elixir
    - https://github.com/dnsbl-io/chaosmonkey-elixir
    - https://github.com/meadsteve/chaos-spawn
  - ErlangVM
    - https://www.youtube.com/watch?v=W_j-5BeiaYE
    - http://www.lambdadays.org/static/upload/media/1519982459833986szymonmentelsharableorchestratingmyhem.pdf
  - Google Cloud Platform
    - https://blog.spinnaker.io/running-chaos-monkey-on-spinnaker-google-compute-engine-gce-155dc52f20ef
  - Hadoop
    - https://medium.com/@tanmay.avinash.deshpande/business-continuity-plan-disaster-recovery-is-too-old-make-a-way-for-chaos-engineering-bc9b3e9f5ebd
  - Java
    - https://codecentric.github.io/chaos-monkey-spring-boot/
  - Kafka
    - https://kafka.apache.org/
    - https://sookocheff.com/post/kafka/kafka-in-a-nutshell/
    - https://www.youtube.com/watch?v=YNs2fVxYzq0
    - https://medium.com/@tanmay.avinash.deshpande/business-continuity-plan-disaster-recovery-is-too-old-make-a-way-for-chaos-engineering-bc9b3e9f5ebd
  - Kubernetes
    - https://github.com/asobti/kube-monkey
    - https://medium.com/chaosiq/exploring-multi-level-weaknesses-using-automated-chaos-experiments-aa30f0605ce
    - https://medium.com/chaosiq/improve-your-cloud-native-devops-flow-with-chaos-engineering-dc32836c2d9a
    - https://github.com/target/pod-reaper
    - https://github.com/bloomberg/powerfulseal
  - OpenShift
    - https://github.com/Produban/monkey-ops
    - https://www.openshift.com/
  - Private Cloud
    - https://github.com/outbrain/GomJabbar
    - https://www.outbrain.com/techblog/2017/06/failure-testing-for-your-private-cloud-introducing-gomjabbar/

    - https://next.nutanix.com/blog-40/chaos-monkey-for-the-enterprise-cloud-27781
  - Scala
    - https://www.cakesolutions.net/teamblogs/chaos-experiment-split-braining-akka-clusters
    - https://github.com/mesosphere/chaos
  - Spark
    - https://medium.com/netflix-techblog/can-spark-streaming-survive-chaos-monkey-b66482c4924a
  - Spring Boot
    - https://codecentric.github.io/chaos-monkey-spring-boot/
  - TCP
    - https://github.com/shopify/toxiproxy
  - VMWare
    - https://www.youtube.com/watch?v=R430b9JhyoI
    - https://github.com/strepsirrhini-army/chaos-lemur
    - http://www.thecloudcast.net/2017/05/the-cloudcast-299-discipline-of-chaos.html
- Chaos Monkey
  - General
    - https://medium.com/production-ready/chaos-engineering-101-1103059fae44
    - https://medium.com/production-ready/chaos-monkey-for-fun-and-profit-87e2f343db31
  - Deployment
    - https://netflix.github.io/chaosmonkey/How-to-deploy/
  - Papers
    - https://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p371.pdf
  - Scenarios/Use Cases
    - https://www.ibm.com/developerworks/library/a-devops4/index.html
    - https://www.ibm.com/blogs/bluemix/2015/12/resilience-testing-insights-from-the-pros/
    - https://developer.ibm.com/recipes/tutorials/chaos-monkey-on-ibm-cloud-private-simple-script-to-randomly-kill-your-k8-pods/
    - https://www.lifehacker.com.au/2014/04/launch-bees-and-monkeys-5-cloud-lessons-nab-learned-from-aws/
    - https://blog.codinghorror.com/working-with-the-chaos-monkey/
    - https://www.battery.com/powered/youve-heard-of-the-netflix-chaos-monkey-we-propose-for-cyber-security-an-infected-monkey/
    - https://www.baeldung.com/spring-boot-chaos-monkey
    - https://medium.com/production-ready/using-chaos-monkey-whenever-you-feel-like-it-e5fe31257a07
    - https://blog.serverdensity.com/building-chaos-monkey/
- Netflix
  - FIT
    - https://medium.com/netflix-techblog/fit-failure-injection-testing-35d8e2a9bb2
  - https://www.infoq.com/news/2018/07/chaos-eng-netflix
  - https://medium.com/netflix-techblog/chaos-engineering-upgraded-878d341f15fa
  - https://medium.com/netflix-techblog/automated-failure-testing-86c1b8bc841f
  - https://medium.com/netflix-techblog/chap-chaos-automation-platform-53e6d528371f
  - https://www.infoq.com/presentations/netflix-chaos-platform
- Simian Army

- General
  - https://github.com/Netflix/SimianArmy
  - https://github.com/Netflix/SimianArmy/wiki/
  - https://medium.com/netflix-techblog/the-netflix-simian-army-16e57fbab116
- Failure/Tooling
  - http://blog.hut8labs.com/gorillas-before-monkeys.html
  - https://www.slideshare.net/InfoQ/resiliency-through-failure-netflixs-approach-to-extreme-availability-in-the-cloud
- Strategies
  - https://github.com/Netflix/SimianArmy/wiki/The-Chaos-Monkey-Army
- Variants
  - https://en.wikipedia.org/wiki/Chaos_Monkey#Chaos_Monkey