chaos-course

unknown

авг. 11, 2022

Оглавление

ГЛАВА 1

Структура курса

1.1 День 0. Хаос-тестирование: нужно ли вам, готовы ли вы?

1.2 День 1. Процессы и роли.

```
Что такое
История с хронолоией
Текущий статус по Infoq
Связь DevOps - SRE - Chaos Engineering
Кто такой Chaos Engineer, компетенции.
Формула ROI на примере инцидентов
Примеры инцидентов и примеры атак
Пример возможного процесса, разбор каждого этапа
Как возможно запустить CE в компании
Измерение эффективности прохождения командами испытаний
```

1.3 День 2. Типы систем и виды атак. Работа с утилитами (практика).

```
Legacy системы:
  Возможная модель атак
          Linux:
          Stress-ng - CPU, RAM
          Blade - CPU, RAM
          tc - сетевой стек, задержки короткие, длинные, потеря пакетов
          Blade - файловая подсистема (забор дескриптеров файлов), быстрая чтение и␣
↪запись
```

(continues on next page)

(продолжение с предыдущей страницы)

```
          Windows:
Cloud Native системы
  Cloud Native Foundation
  Возможная модель атак
          K8S:
          Blade - Kill pods, Force Kill, CPU, RAM вподе
          Chaos Tools -
```

1.4 День 3. Автоматизация (практика).

```
  Возможные этапы автоматизации, зрелось
  Варианты готовых платформ с Foundation
  Jenkins + Groovie + Инструмент
  Ключи?
  Минусы решения.
```

ГЛАВА 2

Результаты обучения

Обсуждение документа

Документ обуждается тут. Видеокомментарий:

2.1 Основные понятия и темы

2.1.1 1. Надежность традиционных и распределенных системы

Надежность:

- важность для пользователей
- требования и SLA
- метрики и мониторинг

Особенности распределенных систем:

- сложно спрогнозировать влияние большого числа случайных внешних факторов
- скрытое внутреннее состояние
- работает на стенде, не работает на проде
- по частям все работает, а вместе нет
- самовосстановление как желаемое поведение

Основные идеи:

1. Сложность и масштаб информационных систем растет
2. Поведение распределенных систем может сильно отличаться от ожидаемого
3. Надежность нужно проверять:

- в окружении и конфигурациях, близких к проду -> выбор стенда
- с нагрузкой, близкой к реальной -> выбор стенда
- с различными ошибками и событиями внешней среды (самыми разрушительными или самыми частыми, уже наблюдаемыми или предполагаемыми) -> дизайн chaos-испытаний
- со сложными сценариями, отражающими практику работы распределенных систем -> дизайн chaos-испытаний
- используя релевантный инструментарий -> выбор инструментов
- делать это часто, на систематической основе -> автоматизация

2.1.2 2. Как начать испытания

Разовое chaos-испытание:

- Логика эксперимента:
  - "устойчивое состояние"
  - "гипотеза" (точка отказа, условия и частота возникновения, характер сбоя)
  - способ воспроизвести и увидеть отказ
  - как использовать результаты
- Подготовка и проведение испытания (стенд, мониторинг, нагрузка, выбор инструментов)
- Анализ и использование результатов

2.1.3 Как сделать хаос-испытания устойчивой практикой

Систематические chaos-испытания:

Условия:

- Важность надежности для бизнеса и клиентов компании
- Техническая зрелость (DevOps/SRE)
- Возможность дорабатывать системы (есть кому, есть мотивация)

Мероприятия:

- Пробные разовые испытания, их обсуждение
- Систематизация опыта разовых chaos-испытаний, инициативные изменения в процессах работы команд
- Автоматизация и масштабирование chaos-испытаний, обязательные требования ко всем командам
- ROI хаос-испытаний

Люди и команды:

- Требуемая квалификация chaos-инженера
- Варианты организации chaos-испытаний в компании

2.2 Знания

Слушатель курса получит следующие знания и практические навыки.

- Как строить процесс Chaos Engineering
- Какие навыки важны для хаос-инженера
- Получит представление об инцидентах и зависимостях **ЧЕГО?** от экспериментов
- Научится считать ROI испытаний **(навык, а не знания)**
- Получит представление об автоматизации хаос-испытаний

2.3 Практические умения и навыки

- Сможет простроить процесс ЧЕГО?, проаудировать процесс ЧЕГО?
- Сможет провести оценку необходимости проведения испытаний в конкретной системе
- Сможет составить матрицу испытаний системы
- Сможет провести испытания систем, базирующихся на linux, Windows, Kubernetes
- Сможет использовать популярные инструменты - tc, stress-ng, blade, blade-operator, chaosblade
- Сможет простроить автоматизацию Jenkins + groovie

ГЛАВА 3

Глоссарий

Chaos engineering (chaos-инжиниринг, chaos-инженерия)
: Дисциплина и набор практик по проведению экспериментов, подтверждающих способность распределенных информационных систем противостоять неблагоприятным условиям в эксплуатационной среде. (адаптировано из **Principles of Chaos**).

DevOps

1. Все самое лучшее в процессах разработки и эксплуатации информационных систем, особенно в глазах бизнес-партнеров и кадровых служб.
2. Современное название работы системных администраторов.
3. Отлаженный релизный цикл ПО на конкретной инфраструктуре.

Injection, fault injection (test method)
: to be added

Observability (наблюдаемость)

1. Свойство продвинутых систем мониторинга, по мнению их вендоров.
2. В теории управления – возможность выявить истинное внутреннее состояние системы по измеримым внешним данным.

Site Reliability Engineering (SRE)
: Улучшенный DevOps, признающий наличие сбоев и проблем с надежностью информационных систем. Концепция придумана и популяризируется компанией Google. **Ссылка на книгу.**

Автоматизация испытаний
: Методики подготовки и проведения испытаний, позволяющие включить chaos-тесты в релизный цикл программного обеспечения (ПО) на инфраструктуре и условиях, близких к эксплуатационой среде, и обеспечивающие масштабирование тестирования надежности и отказустойчивости.

Атака
: Плановое создание неблагоприятных условий работы системы в ходе chaos-эксперимента. См.также "Гремлин".

Замечание: термин чаще используется в испытаниях безопасности, где "атака" повторяет действия злоумышленника, пытающегося получить доступ к системе. В испытаниях надежности термин используется обезличено – никто ни на кого не нападает, атака воспроизводит условия, приводящие к сбою, причем эти условия в процессе эксплуатации обычно возникают стихийно, без участия какого-либо злоумышленника.

Гипотеза
Из чего состоит?

Деградация
Существенное замедление или частичная потеря работоспособности системы (ссылка), часто происходящее без явного отказа (ссылка).

Испытание, chaos-испытание
См. эксперимент (ссылка)

Мониторинг
Добавить описание

Отказ
Синонимы: "дизастр" (disaster), инцидент, сбой. См. деградация (ссылка).

ПО
Программное обеспечение.

Релизный цикл (release cycle)
Процесс разработки и эксплуатации ПО, применяемый конкретной компанией или командой.

Система, автоматизированная система (АС), сервис
Конкретная система, которую мы тестируем.

Тест, chaos-тест
См. эксперимент (ссылка)

Точка отказа (failure point)
Добавить определение

Уязвимость (vulnerability)
Потенциальная точка отказа (ссылка).

Эксперимент, chaos-эксперимент
Контролируемое внесение изменений в условия работы сервиса на тестовом стенде или в эксплуатационной среде. Эксперимент планируется для подтверждения или опровержения заранее сформулированной гипотезы относительно поведения системы в неблагоприятных условиях. Отрицательные результаты эксперимента, показывающие, что система не справляется должным образом с неблагоприятными условиями, являются основанием для последующей доработки системы. Результаты доработки системы, в свою очередь, проверяются повторным экспериментом. Синонимы: chaos-испытание, chaos-тест.

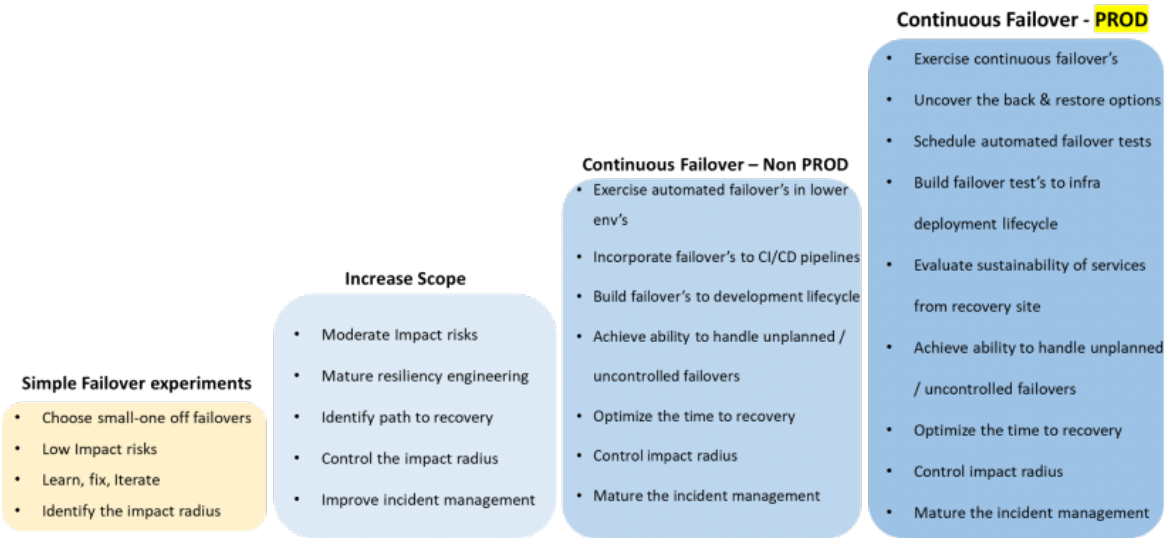
ГЛАВА 4

Ссылки и литература

4.1 Блоги

- What is Chaos Engineering by Gary Parker. February 23, 2022
- A Practical Guide to Chaos Engineering. Cigniti..

  Note maturity model:



- Ронжин П., Казаков В. Надежность, отказоустойчивость, доступность. Синонимы или?..09 марта 2015.

4.2 Организации

- Cloud Native Computing Foundation (CNCF)
- Gremlin