

Полная программа

SRE практики и инструменты

Site Reliability Engineering, как дисциплина, решает проблемы надежности и доступности сервисов

Длительность курса: 114 часов

Модуль 1. Введение в SRE

Тема 1
Введение в SRE // ДЗ

Цель занятия
обсудить историю SRE, связь с DevOps; рассмотреть варианты реализации сервисами и различные варианты Reliability Engineering: Site, Production, Database.

Домашние задания

Настройка окружения

Цель
Нужно подготовить окружение для дальнейшей работы по курсу VSCode + MAMP (либо другая IDE/редактор) Docker Desktop (либо Docker + minikube) <https://www.docker.com/products/docker-desktop> WSL2 Ubuntu 20.04 (либо другой Linux) <https://docs.microsoft.com/en-us/windows/wsl/install-win10> Git Vagrant <https://www.vagrantup.com/> настроенный libvirt или virtualbox <https://www.virtualbox.org/> Ansible Molecule <https://molecule.readthedocs.io/en/latest/> Terraform <https://www.terraform.io/> Дополнительное ПО <https://www.mkdocs.org/> <https://diagrams.mingrammer.com/docs/guides/diagram>

Модуль 2. Everything as a Code

Тема 1
GIT 101: совместная работа, CI, вендоринг

Цель занятия
разобрать основные приемы работы и обсудить среду эксплуатации с точки зрения SRE.

Краткое содержание

типичная для SRE инженерная культура; цикл Знать - Делать - Обучаться; измеряем все; примеры первых метрик; примеры первых шагов по внедрению и утилиты; риски внедрения и антипаттерны.

Тема 2
Основные принципы SRE // ДЗ

Цель занятия
обсудить управление конфигурацией с точки зрения SRE; рассмотреть различные подходы. IaC Ansible Clouds

Домашние задания

Дописать роль для исполнения на другом дистрибутиве (.deb/.rpm)
Цель
Ознакомиться со структурой Ansible role Освоить основные техники работы с кодом для Ansible

Задание
Выберите одну из ролей ниже или используйте любую другую роль: <https://github.com/kevint/ansible-role-monica> <https://github.com/kevint/ansible-role-dedcon> <https://github.com/kevint/ansible-role-funkwhale>
Форкните код Допишите необходимый для исполнения на другом дистрибутиве код (например на CentOS/Rhel/AWS linux)

Тема 3
Практика управление конфигурацией. Terraform // ДЗ

Модуль 3. Экосистема Kubernetes

Тема 1
Linux 101: cgroups/namespaces/network/containers

Тема 2
Экосистема Kubernetes

Тема 3
Практика управление конфигурацией. Helm

Тема 4
QA-сессия

Цель занятия
синхронизироваться по вопросам учёбы.

Краткое содержание
обратная связь; ответы на вопросы.

Тема 5
SLI, SLA, SLO и управление рисками // ДЗ

Цель занятия
управление рисками - одна из самых важных практик. Обсудим, как SRE оценивает сервис для того, чтобы объективно принимать решения. SLI, SLA, SLO - фундаментальные понятия для SRE. Рассмотрим каждый из этих понятий и определим показатели для сервиса.

Краткое содержание
SLI; SLA; SLO; error budget.

Домашние задания

Сформулировать SLO для тестового окружения
Цель
Закрепить понимание SLO/SLI Результат: Получить документ, описывающий SLO для тестового окружения из предыдущей домашней работы

Задание
Определите SLI для приложения <https://github.com/Zenahr/flask-sqlite3-todo-crud> Определите SLI для окружения Определите SLO для окружения

Модуль 4. Автоматизация (Toil elimination)

Тема 1
Автоматизация: Low-Code/No-Code

Тема 2
Автоматизация: CLI, Data transformation // ДЗ

Цель занятия
рассмотреть подход SRE автоматизации, а также примеры ее реализации - как успешные, так и неудачные.

Краткое содержание
Shell/JQ.

Домашние задания

Расширить вывод API и отдать его в формате Prometheus
Цель
Использовать базовую автоматизацию сбора данных и их конвертацию в нужный формат, по возможности использовать утилиты из лекции

Задание
Формат экспортера можно найти тут https://github.com/prometheus/docs/blob/master/content/docs/instrumenting/exposition_formats.md#text-for-mat-example
Варианты:
1. Приложение можно найти тут <https://github.com/Zenahr/flask-sqlite3-todo-crud/blob/master/app.py> отвечает на три запроса: / GET /add POST /update POST Проверьте код ответа приложения, время ответа либо иные важные для ваших SLI на данные запросы и отдайте эту информацию в формате Prometheus
2. Получите ответ от API <http://open-notify.org/Open-Notify-API/ISS-Locate-Now/> и сконвертируйте его в формат Prometheus

Тема 3
Автоматизация: Runbook Automation (Jenkins/AWX/Rundeck) // ДЗ

Цель занятия
лямбды/ift

Домашние задания

Напишите workflow для nbn
Цель
Освоить на практике работу с workflow Использовать навыки работы с структурированным текстом

Задание
Нужно проверить что информация в CSV и JSON идентична. Для этого создать nbn workflow сравнивающий два источника данных
URL: <http://xmlcalendar.ru/data/ru/2022/calendar.csv> <http://xmlcalendar.ru/data/ru/2022/calendar.json>
Добавить workflow в ваш git repo и опубликовать в Readme ссылку по которой можно импортировать workflow в nbn

Модуль 5. Мониторинг и практика оповещений

Тема 1
Задания мониторинга и алертинга

Тема 2
Observability

Тема 3
QA-сессия

Цель занятия
синхронизироваться по вопросам учёбы.

Краткое содержание
обратная связь; ответы на вопросы.

Модуль 6. Практики SRE

Тема 1
Непрерывная поставка и управление изменениями // ДЗ

Цель занятия
обсудить, как обеспечить уверенность в стабильности и качестве выпускаемого продукта с помощью непрерывной поставки.

Домашние задания

В репозитории на гитлаб/битбакет/github actions сделать автоматический трюг тестов на каждый коммит
Цель
Цель - самостоятельно настроить процесс CI/CD.

Задание
Для репозитория, созданного во втором домашнем задании, (или любого форкнутого проекта с гитхаба) необходимо настроить процесс CI/CD.
Использовать инструменты CI/CD, например, <https://github.com/marketplace/circleci> <https://about.gitlab.com/stages-devops-lifecycle/continuous-integration/> настроить автоматическую запуск пайплайна по комиту (у обоих этих инструментов есть бесплатный пайплайн период, или можно поднять локальный гитлаб в докере, например, <https://docs.githab.com/ee/install/docker.html>).

В пайплайн должны входить:

- сборка проекта (желательно собирать приложение в Docker-контейнер, пригодится в следующем дз. Имеет смысл в качестве базового образа брать Linux alpine https://hub.docker.com/_/alpine/, тк это минимизированный образ, в котором нет ничего лишнего. За счет выбора базового имаджа можно существенно сократить время сборки).
- запуск тестов (достаточно наличия одного, самого простого теста. Самое главное - чтобы для запуска тестов была сделана отдельная джоба, которую можно запустить без сборки проекта)

Тема 2
Управление релизами // ДЗ

Цель занятия
рассмотреть практику управления релизами и координацию процессов; разработать чек-лист для запуска; обсудить приемы надежных релизов и роль SRE инженера.

Домашние задания

отбрасываться от мастер-репо, написать свой кусок кода, создать PR и эвержить обратно

Тема 3
Тестирование надежности систем // ДЗ

Цель занятия
рассмотреть виды тестирования ПО, процессы и инструменты; обсудить Chaos Engineering и проведение учений.

Домашние задания

протестировать тестовый сервис из двух микросервисов
Цель
1. написать юнит-тесты к той паре микросервисов 2. написать контракт (и проверить его) 3*. настроить отказоустойчивую балансировку инстансов одного из сервиса и устроить chaos

Тема 4
Управление нагрузкой и перегрузок и сбоев // ДЗ

Цель занятия
обсудить балансировку нагрузки на уровне фронтенда и датацентра; рассмотреть политики балансировки нагрузки.

Домашние задания

Проверить на практике алгоритмы распределения нагрузки
Цель
Закрепить понимание алгоритмов распределения нагрузки
Обучиться работе с Jupyter для визуализации и симуляции процессов

Задание
1.Сконируйте репозиторий <https://github.com/jolynch/performance-analysis>
2. Запустите симуляцию очередей https://github.com/jolynch/performance-analysis/blob/master/notebooks/queueing_theory/queueing_analysis.ipynb
3. Запустите симуляцию Load Balancer https://github.com/jolynch/performance-analysis/blob/master/notebooks/queueing_theory/load_balancing_analysis.ipynb

Тема 5
Практика on-call и жизненный цикл SRE команды

Цель занятия
погрузиться в жизнь дежурного инженера; поговорить про организацию и культуру дежурств; разобрать реализации на практике.

Тема 6
Практика постмортемов

Цель занятия
поговорить про философию постмортемов; рассмотреть хороший и плохой постмортем; понять, с чего начать внедрение культуры постмортема; разобрать примеры, инструменты и шаблоны.

Тема 7
Практика диагностики и решения проблем

Цель занятия
рассмотреть примеры анализа реальных ситуаций и изучить инструментарий.

Тема 8
Customer Reliability Engineering

Модуль 7. Поиск работы

Тема 1
Как подсветить навыки в резюме. Шаблон резюме

Тема 2
Этапы собеседования: чего ждать? Live-собеседование с поведенческими вопросами

Тема 3
QA-сессия

Тема 4
Dashboard as a Code

Модуль 8. Проектная работа

Тема 1
Выбор темы и организация проектной работы

Цель занятия
выбрать и обсудить тему проектной работы; спланировать работу над проектом, ознакомиться с регламентом работы над проектом.

Краткое содержание

правила работы над проектом и специфика проведения итоговой защиты; требования к результату проекта и итоговой документации.

Тема 2
Консультация по проектам и домашним заданиям - промежуточная приемка

Цель занятия
получить ответы на вопросы по проекту, ДЗ по курсу.

Краткое содержание

вопросы по улучшению и оптимизации работы над проектом; затруднения при выполнении ДЗ; вопросы по программе.

Тема 3
Защита проектных работ

Цель занятия
защитить проект и получить рекомендации экспертов.

Краткое содержание

презентация проектов перед комиссией; вопросы и комментарии по проекту.

Домашние задания

Итоговый проект

Цель
В первый SRE в компании, которая построена вокруг блага, продающего винно, index.cloud балансировка nginx wordpress mysql memcached nfs очередь для обработки заказов и ml по логам формирующей ленту контента пользователю.

Задание
Внедряем:
Первый этап Запустить DevOps CI/CD для вашего кода Automate infrastructure scaling Как вы будете соблюдать правило 50/50 (стратегия) Проведите SRE тренинг для других команд (pdf презентация)

Второй этап Определите SLA,SLO,SLI. Чем вы руководствовались, с кем вы общались в компании? Измерьте поведение вашей системы на соответствие SL*. Как вы этого добились? Определите Error-Budget (velocity vs quality) Определите OKR для SRE

Третий этап Внедрите мониторинг Golden signals: latency, saturation, traffic, errors Внесите релиза упал memcached. Выпустите Blameles postmortem Исправьте архитектуру чтобы этого больше не произошло

Четвертый этап Проверьте что падение memcached больше не ломает систему проведите эксперимент Handle improvement discussion Response scenarios based on proactive monitoring