



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«Дальневосточный федеральный университет»

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

Кафедра компьютерных систем

Садаев Федор Андреевич

Использование docker контейнеров для автоматизации тестирования

КУРСОВАЯ РАБОТА

по направлению подготовки 09.04.02 Информационные системы и
технологии, магистерская программа «Большие данные и облачные
технологии»

Студент гр. М8118 _____
подпись

Руководитель к.ф.-м.н.
_____ Ю. А. Шевченко

Регистрационный № _____

Оценка _____

_____ И.О. Фамилия
подпись

_____ И.О. Фамилия
подпись

«_____» _____ 2019 г.

«_____» _____ 2019 г.

**г. Владивосток
2019**

Оглавление

Введение	3
1 Автоматизированное тестирование ПО	5
1.1 Понятие тестирования.....	5
1.2 Применение автоматизированного тестирования	5
2 Инструменты для автоматизации тестирования.....	7
2.1 Selenium	7
2.2 Katalon Studio.....	8
3 Использование docker	9
3.1 Docker.....	9
3.2 Пример реализации	9
Заключение	12
Список литературы	13

Введение

В настоящее время тестирование занимает одну из важнейших ролей в процессе разработки ПО. Ни один крупный продукт не может обходиться без тестирования и зачастую тестирование и отладка функционала занимает гораздо больше времени, чем сама разработка этого же функционала.

Но при разработке ПО время всегда ограничено и тестировать функционал должным образом может просто не хватать времени. Большую часть времени в тестировании ПО занимает регрессионное тестирование продукта. При регрессионном тестировании тестировщику необходимо проверить весь функционал, который уже протестирован и внедрён в продукт. И логично, чем больше продукт, тем больше времени занимает регрессионное тестирование.

Чтобы частично освободить тестировщиков от проведения некоторых тестов используется автоматизация тестирования. Она позволяет переложить часть тест кейсов на автоматизированную проверку, тем самым освобождая время тестировщиков на более важные тесты.

Но в больших проектах возникает проблема в масштабируемости системы авто тестирования, так как в больших проектах количество тестов исчисляется тысячами возникает вопрос в параллельном исполнении тест кейсов. Одним из решений данной проблемы является использование контейнеризации.

Цель этой работы - показать способ автоматизации тестирования веб приложения с использованием docker контейнеров.

Объектом является веб приложение. Предметом является система автоматизированного тестирования.

Задачи работы:

- Описать понятие тестирования.
- Описать какие виды тестирования автоматизируются.
- Рассмотреть инструменты для автоматизации тестирования

- Привести пример использования docker для организации автоматизированного тестирования.

1 Автоматизированное тестирование ПО

1.1 Понятие тестирования

Под тестированием принято понимать деятельность, выполняемую для оценки и улучшения качества ПО. В общем случае тестирование базируется на обнаружении дефектов и проблем в программных системах.

Автоматизированное тестирование ПО — процесс тестирования программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, производятся автоматически с помощью инструментов для автоматизированного тестирования.

В свою очередь, инструмент для автоматизированного тестирования — это программное обеспечение, посредством которого осуществляется создание, отладка, выполнение и анализ результатов прогона тест-кейсов.

Тестирование программных систем состоит из динамической верификации поведения программ на конечном наборе тестов. При этом тесты выбираются из обычно выполняемых действий прикладной области и обеспечивают проверку соответствия ожидаемому поведению системы.

1.2 Применение автоматизированного тестирования

Первым пунктом в этом списке стоит тестирование производительности. Нагрузочное, стрессоустойчивое, тестирование на стабильность... Без автоматизации его выполнение трудно себе представить. По этой причине имеется широкий выбор продуктов от разных производителей и столь же высокие цены, даже в случае неудобного и слабо функционального инструмента.

Следом идёт регрессионное тестирование. Означает оно проверку ПО на корректность функциональности, выпущенной и протестированной в предыдущей версии. Выполняется с регулярной частотой, задаваемой в зависимости от условий: у кого-то с каждым новым билдом, а у кого-то с каждой версией для заказчика.

Конфигурационное тестирование – выполнение одних и тех же тестов в разных условиях. То есть, когда один или несколько компонентов архитектуры системы требуется проверить в разном окружении, обычно заявленном в изначальных требованиях. Например: поддержка СУБД от разных производителей, работа в разных клиентских браузерах, использование в нескольких ОС и т.п. То есть некий аналог регрессионного тестирования, но в рамках одной версии системы.

Функциональное тестирование. Ясно, что здесь речь идёт о проверке нового функционала. Иногда бывает, что без автоматизации никак не обойтись. Даже если нужно выполнить тестирование только один раз. Обычно, впоследствии эти тесты и используются для регресса.

Установочное тестирование, выполняется для проверки условий инсталляции (и настройки) продукта с учётом тех или иных требований к системе от заказчика.

2 Инструменты для автоматизации тестирования

2.1 Selenium

Selenium — это инструмент для автоматизации действий веб-браузера. В большинстве случаев используется для тестирования Web-приложений, но этим не ограничивается.

Считается, что Selenium является самым популярным фреймворком с открытым исходным кодом, предназначенным для автоматизации тестирования веб-приложений. Разработанный где-то в далеких двухтысячных и развивающийся в течение последующего десятилетия, Selenium представляют собой фреймворк для автоматизации процесса тестирования веб-приложений. За это время он успел завоевать сердца многих тестировщиков, особенно тех, у кого в распоряжении имеются продвинутые навыки программирования и опыт написания скриптов. Selenium можно рассматривать в качестве родоначальника некоторых современных инструментов автоматизации тестирования с открытым исходным кодом, например: Katalon Studio, Watir, Protractor и Robot Framework.

Фреймворк Selenium поддерживается несколькими ОС (Windows, Mac, Linux), а также многими браузерами (Chrome, Firefox, IE, и браузерами Headless). Скрипты для данного фрейма можно написать на большинстве популярных сегодня ЯП: Java, Groovy, Python, C#, PHP, Ruby и Perl.

Однако стоит отметить, что данный фреймворк имеет как плюсы, так и минусы. К преимуществам можно отнести гибкость, а также возможность написания сложных и эффективных скриптов для тестирования разрабатываемых приложений. С другой стороны, для того чтобы начать работать с Selenium, тестировщик должен обладать незаурядными знаниями в программировании и быть готовым уделять некоторое количество своего времени и энергии для написания специальных фреймов и библиотек, обеспечивающих выполнение определенных функций в процессе тестирования.

2.2 Katalon Studio

Katalon Studio — это эффективный инструмент для автоматизации процесса тестирования веб-приложений, мобильных приложений и веб-сервисов. Katalon Studio является потомком таких фреймворков, как Selenium и Appium. Он перенял у последних множество преимуществ, связанных с интегрированной автоматизацией тестирования ПО.

Для начала работы с данным инструментом вы можете как обладать начальными знаниями в тестировании ПО, так и быть настоящим гуру своего дела. Люди, далекие от программирования, могут с легкостью запустить свой проект по автоматизации тестирования (например, запустив функцию Object Spy для записи тестовых скриптов), а для программистов и опытных тестировщиков Katalon Studio окажется полезным с точки зрения экономии времени при написании новых библиотек и поддержке существующих скриптов.

Katalon Studio может быть интегрирован в CI/CD, он прекрасно работает в связке с популярными инструментами во время тестирования ПО: qTest, JIRA, Jenkins и Git. Для него предусмотрена приятная функция — Katalon Analytics, благодаря которой пользователи получают полное представление о процессе тестирования. Для этого предусмотрены специальные отчеты, которые выводятся на экран пользователей в виде метрики, диаграмм и графиков.

3 Использование docker

3.1 Docker

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесён на любую Linux-систему с поддержкой cgroups в ядре, а также предоставляет среду по управлению контейнерами. Изначально использовал возможности LXC, с 2015 года применял собственную библиотеку, абстрагирующую виртуализационные возможности ядра Linux — libcontainer. С появлением Open Container Initiative начался переход от монолитной к модульной архитектуре.

Используя docker контейнеры можно создать удобную среду для автоматизированного тестирования ПО, которую можно развернуть на любой машине с docker. При этом docker позволяет развернуть несколько копий проекта на одной или нескольких машинах для параллельного исполнения тестов. В таком случае масштабируемость системы автотестирования ограничивается только вычислительными мощностями.

Для наиболее распространённых инструментов для автотестирования в docker hub имеются уже готовые к работе образы контейнеров, которые можно использовать в своих проектах.

Например, есть множество контейнеров для selenium, но основные: selenium/standalone-chrome и selenium/standalone-firefox. Они разворачивают selenium сервер для chrome и firefox. После этого на клиентах можно свободно подключаться к контейнеру и запускать тесты без необходимости самостоятельно настраивать сервер на основной ОС.

3.2 Пример реализации

Суть реализации заключается в том, чтобы создать окружение для проведения автоматизированного тестирования с возможностью переноса на

другую машину с минимальными затратами и возможностью масштабирования.

Допустим для написания авто тестов используется комбинация selenium+python+pytest. Это наиболее известная комбинация для реализации автоматизированного тестирования. Но может использоваться и другой язык программирования для работы с selenium или использоваться библиотека unittest вместо pytest. Комбинаций для организации самой архитектуры автоматизированного тестирования вокруг selenium большое множество.

После определения каркаса авто тестов их необходимо развернуть в Docker. Для таких случаев есть специальная служба docker-compose которая позволяет описать несколько контейнеров и способ их запуска в файле docker-compose.yml после чего с помощью команды docker-compose up поднять все необходимые сервисы разом.

В предложенной выше комбинации необходимо создать 2 контейнера. Контейнер для selenium доступен из хаба докера, где уже есть готовые для использования контейнеры которые необходимо просто скачать. Примеры контейнеров были приведены в предыдущем пункте.

Второй контейнер — будет непосредственно запускать тесты. Его можно собрать из обычного контейнера с ubuntu с запуском так называемого Dockerfile в котором описаны команды, которые исполняются в контейнере и настраивают сервис в нём. В предложенном случае необходимо описать установку python и подключение рабочей папки с проектом автотестов.

Описав эти 2 контейнера в файле docker-compose.yml командой docker-compose up -d --build докер скачает, соберёт и поднимет оба контейнера готовых к работе и запуску автотестов.

После этого для переноса проекта на другую машину или облако достаточно лишь перекинуть проект с кодом тестов и файлы конфигурации docker и запустить команду и на новой машине будет поднят абсолютно такой же проект.

Таким образом можно поднять так называемый рой контейнеров и настроить каждый контейнер на выполнение своего набора тестов. Таким образом выполняя несколько тестов параллельно.

При этом docker весьма гибок и после прохождения тестов можно выключать уже не нужные контейнеры или же запускать на них следующий набор тестов.

Заключение

В данной работе было рассмотрено понятие автоматизированного тестирования ПО, его применение, пару инструментов, которые используются для автоматизации тестирования и привёл пример реализации проекта автоматизированного тестирования ПО с использованием docker.

А также решены следующие задачи:

- Описаны понятия тестирования.
- Описаны какие виды тестирования автоматизируются.
- Рассмотрены инструменты для автоматизации тестирования
- Приведён пример использования docker для организации автоматизированного тестирования.

Список литературы

1. Автоматизация тестирования программных систем [Электронный ресурс] URL: <https://habr.com/ru/post/160257/>
2. Docker. Зачем и как [Электронный ресурс] URL: <https://habr.com/ru/post/309556/>
3. Тестирование. Фундаментальная теория [Электронный ресурс] URL: <https://habr.com/ru/post/279535/>
4. Docker: Окружение для тестирования [Электронный ресурс] URL: <https://habr.com/ru/post/275513/>
5. Docker [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/Docker>
6. Савин Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах.— М.: Дело, 2007. — 312 с.
7. Рекс Блек. Ключевые процессы тестирования - М.: Издательство Лори, 2014. - 544 с.
8. Канер Сэм Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: Пер. с англ./Сэм Канер, Джек Фолк, Енг Кек Нгуен. — К.: Издательство «ДиаСофт», 2001. — 544 с.