



УДК 004.054

Преображенская Тамара Викторовна, ст. преподаватель,
Новосибирский государственный технический университет, г. Новосибирск

Симонов Владислав Игоревич, студент,
Новосибирский государственный технический университет, г. Новосибирск

АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ В СРЕДЕ GITLAB CI/CD AUTOMATED TESTING IN THE GITLAB CI/CD ENVIRONMENT

Аннотация. Опыт освоения среды GitLab - платформы для управления жизненным циклом разработки программного обеспечения и автоматизированного тестирования и пошаговое описание технологии работы в среде полезно при изучении дисциплины Программная инженерия для получения новых ИТ-компетенций.

Abstract. The experience of mastering the GitLab environment - a platform for managing the life cycle of software development and automated testing, and a step-by-step description of the technology of working in the environment is useful when studying the discipline of Software Engineering to gain new IT competencies.

Ключевые слова: Автоматизация тестирования, программная инженерия, методы и технологии разработки программных систем, GitLab, CI/CD

Keywords: Test automation, software engineering, methods and technologies for developing software systems, GitLab, CI/CD

Тестирование – обязательная операция технологического процесса создания программного продукта (приложения) для подтверждения его качества. Тестовый сценарий (ТС) это совокупность инструкций по реализации тестового набора. В



идеальной ситуации, после любого внесённого изменения в программу нужно заново полностью протестировать программу – повторить ТС. В больших и сложных проектах, при создании высоконагруженных приложений, когда ТС регулярно повторяются, очень сложны и трудоемки появляется необходимость использования систем автоматизированного тестирования. Наиболее известные из них - GitLab и Jenkins.

Непрерывная интеграция (Continuous Integration, CI) и непрерывная поставка продукта (Continuous Delivery, CD) представляют собой культуру, набор принципов и практик, позволяющих разработчикам чаще и надежнее развертывать создаваемое программное обеспечение. CI/CD – одна из DevOps-практик, ее относят к agile-практикам (гибким методам). Автоматизация развертывания позволяет разработчикам в большей мере сосредоточиться на реализации бизнес-требований, качестве кода и его безопасности. DevOps (акроним от англ. development & operations) - это подход к взаимодействию разработчиков, тестировщиков и других IT-специалистов в команде.

GitLab и Jenkins — программные системы (ПС), используемые для непрерывной интеграции и непрерывной доставки (CI/CD). Обе они - системы с открытым исходным кодом. GitLab — платформа, включающая в себя контроль версий, управление проектами, непрерывную интеграцию и многое другое. Это единое интегрированное приложение, предоставляющее все инструменты разработчикам для планирования, разработки, тестирования и доставки программного обеспечения. В GitLab интегрируются различные инструменты для предоставления комплексного решения CI/CD.

Jenkins — это сервер непрерывной интеграции, предназначенный для автоматизации процесса создания, тестирования и развертывания программного обеспечения. Его можно использовать для автоматизации различных задач, включая сборку, тестирование и развертывание изменений в проекте. Jenkins



можно интегрировать в различные инструменты и платформы, включая GitLab, для предоставления полного решения CI/CD.

GitLab, и Jenkins — популярные инструменты для CI/CD, и какой из них лучше для конкретного проекта, зависит от конкретных потребностей и целей проекта. GitLab может быть лучшим выбором для команд, которым нужна полная интегрированная платформа для управления жизненным циклом разработки программного обеспечения, а Jenkins - для команд, которым требуется более гибкое, настраиваемое решение для непрерывной интеграции и доставки.

GitLab CI/CD. Выбор GitLab обусловлен простотой настройки, что важно для первого знакомства с подобными ПС. GitLab - один из любимых DevOps-инженерами инструментов (написан в 2011 году на языке Ruby, позже разработчиками переведён частично на Go, украинскими программистами - Дмитрием Запорожцем и Валерием Сизовым). Систему используют не только как средство совместного управления версиями, но и как средство непрерывного тестирования. Рассмотрим пример: несколько разработчиков, работают над одним проектом в разных ветках (версиях) этого проекта, и каждый разработчик вносит свои изменения в проект. Система GitLab при внесении любого изменения будет запускать автоматическое тестирование, в том числе и изменения тестов.

Требования для работы с инструментом:

- Включить виртуализацию в BIOS [1];
- Включить компонент Open SSH Client, он нужен для генерации SSH ключей [2];
- Скачать и установить Git, он нужен чтобы переключаться между версиями проекта [3];
- Скачать и установить Docker, он нужен чтобы автоматическое тестирование могло запускаться на локальном компьютере [4].



Настройка GitLab CI/CD: Необходимо зарегистрироваться в системе GitLab [5], создать хранилище, в хранилище создать пустой репозиторий (Р). Чтобы работать с Р, нужно сгенерировать SSH ключ (об этом говорит предупреждение). Чтобы добавить ключ, нужно нажать на кнопку «Add SHH key». Ключ нужно сгенерировать через компонент Open SSH Client. Чтобы это сделать необходимо открыть командную строку и прописать там «ssh-keygen -t rsa -b 2048». Чтобы найти публичный ключ нужно перейти в папку C:\Users\ИМЯ_ПОЛЬЗОВАТЕЛЯ\.ssh (если папка не видна, то включить показ скрытых элементов). В файле с расширением .pub лежит публичный ключ. Содержимое этого файла копируется (можно открыть блокнотом) и вставляется в соответствующее поле GitLab. Теперь система GitLab позволяет общаться компьютеру с созданным Р. Чтобы связать проект с Р GitLab, необходимо инициализировать Р в папке с проектом. Для этого в командной строке нужно перейти в папку с проектом и написать команду «git init». Чтобы сохранить и отправить изменения необходимо написать последовательно «git add .», «git commit -m “Add source of project”», «git push -u origin main». После отправки изменений на GitLab на рис. 1 видно, что тестирование провалилось из-за неподтверждённого аккаунта.

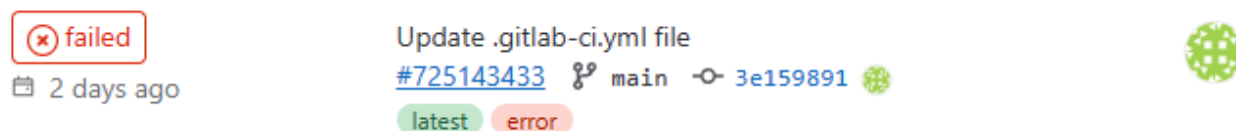


Рис.1 - Проваленный тест

Так как на момент написания этой работы нет возможности добавить банковскую карту российского банка и тем самым подтвердить аккаунт, то необходимо пойти по обходному пути. Обходной путь заключается в установке



компонента, запускающего тестирование, к себе на компьютер, и является абсолютно законным, так как мы пользуемся компонентами и инструкциями, которые предоставляет сервис GitLab. Перед установкой компонента нужно узнать токен для него. Токен можно узнать, перейдя по ссылке «https://gitlab.com/ИМЯ_ХРАНИЛИЩА/ИМЯ_РЕПОЗИТОРИЯ/-/settings/ci_cd», он нужен исключительно для того, чтобы связать тестировщик с системой GitLab и не связан с операциями шифрования. Скачивается бинарный файл компонента: для 64 битной версии, либо для 32 битной версии [6]. В папку по адресу C:\GitLab-Runner (если нет, то нужно её создать) необходимо переместить бинарный файл. В ней, от имени администратора, нужно открыть командную строку и последовательно прописать следующее: «.\ gitlab-runner-windows-amd64 install», «.\ gitlab-runner-windows-amd64 start», «.\ gitlab-runner-windows-amd64 register». Во время регистрации указать «<https://gitlab.com/>» в качестве URL и полученный токен. В самом конце, в качестве исполнителя необходимо написать docker. Теперь при внесении изменений в проект, автоматически запускается тестирование изменений на локальном компьютере, как на рис. 2.

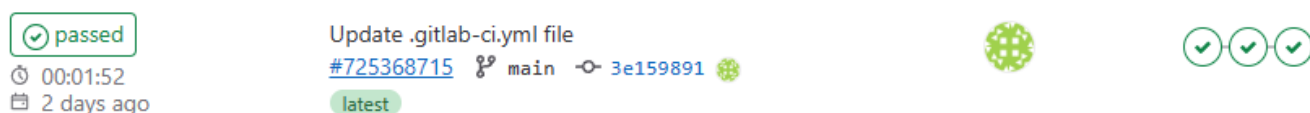


Рис.2 - Пройденный тест

Приведенные сведения могут помочь в создании лабораторных работ для студентов направления Прикладная информатика, например, по дисциплине Программная инженерия и в получении важных навыков для практической работы - новых востребованных ИТ-компетенций на рынке труда.



Список литературы:

1. Как включить аппаратную виртуализацию [Электронный ресурс]: BlueStacks. – Режим доступа: <https://www.support.bluestacks.com/hc/ru/articles/360058102252-Как-включить-аппаратную-виртуализацию-VT-в-Windows-10-для-BlueStacks-5>
2. Установка OpenSSH [Электронный ресурс]: Официальный сайт Microsoft. – Режим доступа: https://learn.microsoft.com/ru-ru/windows-server/administration/openssh/openssh_install_firstuse
3. Downloads Git [Электронный ресурс]: Git. – Режим доступа: <https://git-scm.com/downloads>
4. Install Docker desktop [Электронный ресурс]: Docker. – Режим доступа: <https://www.docker.com/products/docker-desktop/>
5. About GitLab [Электронный ресурс]: GitLab. – Режим доступа: <https://about.gitlab.com/>
6. GitLab Runner Install [Электронный ресурс]: GitLab. – Режим доступа: <https://docs.gitlab.com/runner/install/>