



Jenkins vs TeamCity: лучший инструмент CI в 2024 году?

Описание

Давайте обсудим эти два популярных инструмента непрерывной интеграции и их различия. За последнее десятилетие методика жизненного цикла разработки программного обеспечения претерпела значительные изменения. Раньше для продвижения вперед использовалась водопадная модель, затем появилась agile-методология, в которой разработка велась итеративно, а теперь, в DevOps, и разработка, и операции ведутся итеративно и непрерывно, пока не будет разработано и развернуто полное приложение. Термин CI/CD, который расшифровывается как continuous integration and continuous delivery, стал очень популярным. Это практика DevOps, которая помогает разрабатывать и поставлять приложения гораздо быстрее и надежнее. Это методология, которая автоматизирует все этапы, начиная с бизнес-требований и заканчивая развертыванием на производстве с помощью инструмента CI/CD. Это намного лучше и безопаснее, чем делать все вручную. В настоящее время существует множество инструментов для CI/CD, поэтому выбор подходящего может запутать.

Почему Jenkins ?

Jenkins – самый популярный инструмент непрерывной интеграции с открытым исходным кодом. Он является стандартом де-факто для решений непрерывной интеграции. Вы можете установить Jenkins на основные операционные системы, такие как Windows или Linux, поскольку он работает на Java. Изначально он был создан как инструмент автоматизации сборки для Java-приложений. С тех пор он сильно эволюционировал и имеет более 1400 плагинов, позволяющих легко

интегрироваться с другими платформами и инструментами.



Jenkins

Jenkins – это основной инструмент для непрерывной интеграции и непрерывной доставки в процессе разработки программного обеспечения. Он прост в установке и предоставляет приборную панель для управления проектами, доступ к которой можно легко получить через браузер. Разработчики используют Jenkins для автоматической сборки, интеграции, тестирования и развертывания своего кода. Сначала разработчики проверяют свой код, затем Jenkins подхватывает измененный исходный код, запускает сборку и, если нужно, проводит тестирование. По завершении сборки вы получите результат на приборной панели Jenkins, а разработчику может быть отправлено уведомление, если оно задано.

Почему TeamCity?

TeamCity – это коммерческий CI/CD-сервер, который также основан на java. Это инструмент автоматизации и управления сборкой, созданный компанией JetBrains. Девиз TeamCity – **“Мощная непрерывная интеграция из коробки”**, и этот инструмент его оправдывает. Он предлагает почти все возможности Jenkins с несколькими дополнительными. TeamCity может интегрироваться с Docker для автоматического создания контейнеров с помощью docker-compose. Он поддерживает интеграцию с инструментом Jira для удобного отслеживания проблем. TeamCity поддерживает фреймворк .NET, и вы можете легко интегрировать TeamCity с несколькими IDE, такими как Eclipse, Visual Studio и т. д. Благодаря интеграции с репозиторием артефактов сборки, TeamCity может хранить артефакты в файловой системе сервера TeamCity или во внешнем хранилище. С бесплатной версией TeamCity и лицензией на сервер Professional вы можете создать 100 сборок и 3 агента сборки без каких-либо затрат.

Jenkins против TeamCity

Открытый исходный код против коммерческого

Самое основное отличие заключается в том, что Jenkins – это инструмент непрерывной интеграции с открытым исходным кодом, а TeamCity – коммерческий инструмент. Проект Jenkins выпускается под лицензией MIT и поддерживается разработчиками по всему миру. TeamCity разрабатывается и поддерживается материнской компанией JetBrains.

Интерфейс

Интерфейс TeamCity прекрасен, и он помогает привлечь многих новичков, которые начинают работать с непрерывной интеграцией и ищут инструмент CI. В интерфейсе TeamCity вы можете пометить, комментировать и прикреплять сборки, чтобы организовать их более удобным образом.

Интерфейс Jenkins – это старая школа. Он не очень привлекателен на вид и имеет огромные возможности для улучшения с точки зрения пользовательского интерфейса.

Эксклюзивные возможности

Хотя Jenkins не обладает большим количеством встроенных функций, его огромная экосистема плагинов предоставляет массу возможностей, которые не могут предложить другие инструменты CI. Благодаря таким плагинам Jenkins используется для создания сборок и анализа кода, что помогает улучшить качество кода. С другой стороны, помимо некоторых обычных функций CI-инструмента, TeamCity поставляется с множеством готовых функций. Он поддерживает множество платформ, таких как Java, Ruby, .NET и т. д. Также через TeamCity можно собирать образы докеров. Благодаря поддержке Bugzilla и Jira вы можете с легкостью отслеживать проблемы. Кроме того, сохраняются изменения в сборках и история неудач, что позволяет хранить прошлую информацию о статистике, запускать прошлые сборки и отчеты об истории тестирования.

Установка

Чтобы установить Jenkins в систему, необходимо, чтобы в ней присутствовала Java. Установка Jenkins не представляет сложности, если оба компонента уже присутствуют в системе. После завершения установки вы можете начать работать с Jenkins в его веб-интерфейсе. Установка TeamCity также очень проста. Вам нужно скачать TeamCity Server, перейти к документации и следовать указанным инструкциям.

Интеграции

Мне кажется, что ни один CI-инструмент даже близко не стоит с Jenkins, когда мы говорим о возможностях интеграции. Популярность Jenkins на рынке объясняется тем, что он обеспечивает интеграцию с сотнями бесплатных плагинов. По сравнению с Jenkins, TeamCity может интегрироваться с гораздо меньшим количеством плагинов. Но он предлагает несколько эксклюзивных интеграций, что увеличивает его “возможности из коробки”. Он поддерживает фреймворк .NET и интегрируется с Visual Studio Team Services, обеспечивая интеграцию с облачными провайдерами, такими как AWS, Azure, GCP, VMware.

Безопасность

Положительным моментом коммерческой версии TeamCity является то, что компания JetBrains поддерживает ее, чтобы в случае обнаружения уязвимостей она была исправлена в приоритетном порядке. TeamCity обеспечивает интеграцию с плагином безопасности Snyk, который может выполнять сканирование уязвимостей в конвейере сборки. Он помогает выявить и устранить все риски и угрозы, присутствующие в ваших сборках. Учитывая, что Jenkins – это open-source, снижение рисков может быть отложено, поскольку все зависит от сообщества разработчиков.

Сообщество

Сообщество Jenkins обширнее, чем TeamCity. Поскольку Jenkins имеет открытый исходный код, любой разработчик может внести свой вклад в этот проект. Ежедневно новые разработчики вносят свой вклад в этот проект, чтобы добавить новые функции и сделать его более совершенным инструментом. С другой стороны, TeamCity является коммерческим инструментом и поддерживается командой JetBrains. У него есть профессиональное сообщество поддержки, которое

занимается развитием этого инструмента.

Ценообразование

Jenkins можно использовать бесплатно, так как он имеет открытый исходный код, и именно поэтому он является предпочтительным выбором для многих организаций. Организации экономят значительные средства, не тратясь на CI-инструмент, такой как Jenkins. TeamCity не является бесплатным. Она поставляется с двумя лицензиями- профессиональной и корпоративной. В профессиональной серверной лицензии вы можете использовать 100 конфигураций сборки и 3 агента сборки бесплатно, а после этого – 299 долларов за 1 дополнительный агент сборки и 10 конфигураций сборки. Лицензия корпоративного сервера TeamCity начинается с 3 агентов и предоставляет неограниченное количество конфигураций сборки, начиная с 1 999 долларов.

Сравнительная таблица

	Дженкинс	TeamCity
Написано в	Java	Java
Разработчик	CloudBees, Косукэ Кавагути, свободное программное обеспечение с открытым исходным кодом	JetBrains
Лицензия	Лицензия MIT	Собственное коммерческое программное обеспечение

Характеристики	Открытый источникБогатыйнабор плагиновПростая настройкаПоддержка конвейеров сборкиПлагинрабочего процессаМножество интеграцийВнутреннее размещениеСборка, развертывание или запуск чего-либо asyncИздатель HTMLДроссельная заслонка	Ключевые интеграцииИнтеграция с облакомИстория строительстваПостроить инфраструктуруОтслеживание качества кодаВзаимодействие с VCSОбслуживание системыУправление пользователямиРасширяемость и настройкаПредварительно проверенное обязательство
Простота использования	Да	Да
Пользовательский интерфейс	Очень простой интерфейс	У него привлекательный и красивый интерфейс
Сообщество	Поскольку она с открытым исходным кодом, сообщество больше.	Меньше по сравнению с Дженкинсом
Ценообразование	Бесплатно	Лицензия Professional Server стоит 299 долларов за агента сборки после 3 бесплатных агентов сборки.в то время какСтоимость лицензии Enterprise Server начинается от 1999\$
Интеграции	1400+ интеграций	300+ интеграций
Безопасность	Менее обеспеченный	Более надежно
API	Поддерживает JSON, Python, RESTful API: XML	Поддержка RESTful API с использованием сервисных сообщений и создание плагинов с помощью Open API

Поддержка языков программирования	Java, PHP, C, C++, Fortran, Python и многие другие языки сценариев.	Python, C++, Java, .NET, Ruby, PHP, Node.js и др.
Поддержка контейнеров	Использование плагина Docker для Jenkins	Первоклассная поддержка Docker из коробки
Цикл выпуска	Чаще	Реже
Отчетность	Предоставляет подробные отчеты о сборках и тестах	Предоставляет отчеты с улучшенным веб-интерфейсом

Заключение

Теперь вы знаете, чем отличаются два самых популярных инструмента непрерывной интеграции – Jenkins и TeamCity. Когда вы выбираете инструмент CI для своей организации, вам необходимо проверить несколько параметров, таких как возможности хостинга, доступные интеграции, библиотека многократно используемого кода, поддержка контейнеров, а также простота использования и освоения инструмента. Инструмент непрерывной интеграции, который соответствует этим параметрам, будет отличным вариантом для выбора. Если вы все еще сомневаетесь, попробуйте воспользоваться бесплатными версиями Jenkins и TeamCity. После создания нескольких сборок на этих платформах вам будет легче решить, какой инструмент лучше использовать для непрерывной интеграции ваших проектов.

Дата Создания

14.03.2024