

Механизмы CI/CD

Базовые механизмы конвейера постоянной интеграции и развертывания программных продуктов АО "Национальные квалификации" состоят из следующих технологических этапов:

Тестовая эксплуатация

1. Публикация изменений исходного кода
2. Запуск по расписанию (2 часа ночи) или по триггеру процесса сборки ПО
3. Запуск набора автотестов (юнит-тестирование)
4. Упаковка установочного пакета в формате debian
5. Упаковка установочного пакета в формате tarball+setup.sh
6. Передача собранного deb пакета в репозиторий ПО "Тестовая эксплуатация"
7. Установка или обновление собранного пакета в среде тестовой эксплуатации по расписанию на 5 утра по Московскому часовому поясу или по триггеру.

После успешного завершения всех этапов программное обеспечение доступно в среде тестовой эксплуатации. Если хотя бы один этап завершается с ошибкой, владельцу проекта отправляется отчет с уведомлением о возникшей ошибке.

Опытная и промышленная эксплуатация

1. После успешного тестирования ПО в тестовом контуре тестировщик подписывает Акт тестирования. В Акте фигурирует версия ПО, включающая номер коммита (зафиксированных изменений) и тег версии ПО.
2. После подписания акта тег версии присваивается номеру коммита.
3. Запускается процесс сборки ПО с версией согласно тегу.
4. Формируется журнал изменений из событий между тегами. в журнал попадают только события отмеченные символами +/-/* в начале каждой строки описания изменений.
5. Запускается набор автотестов (юнит-тестирование)
6. Упаковка установочного пакета в формате debian
7. Упаковка установочного пакета в формате tarball+setup.sh
8. Передача собранного deb пакета в репозиторий ПО "Опытная эксплуатация"
9. Отправка уведомлений кураторам и владельцам среды опытной эксплуатации об обновлении ПО включая список изменений.
10. Обновление ПО в среде опытной эксплуатации при помощи агента Puppet
11. После завершения тестирования ПО ведущий разработчик, технический директор или начальник отдела технической поддержки совместно с двумя кураторами регионов подписывают Акт тестирования. Всего устанавливается 3 подписи разными видами уполномоченных лиц.
12. После подписания акта тестирования ПО передается в репозиторий "Промышленная эксплуатация"
13. Отправка уведомлений кураторам и владельцам среды промышленной эксплуатации об обновлении ПО включая список изменений.

Документация

Документация должна автоматически собираться на дату сборки и формирования пакетов в формате PDF с использованием системы справки и поддержки АО НК. Файл документации должен автоматически формировать шапку, подвал, типографику, сноски, версию документа и иные параметры верстки и допечатной подготовки, согласно заданному шаблону.

Сборка программного обеспечения

Для каждого разрабатываемого продукта АО НК должен формироваться Debian пакет программного обеспечения для целей быстрого точечного развертывания, контроля версионности, зависимостей и целостности распространяемого ПО.

Требования к формируемым пакетам:

1. Имя ПО является базовой частью имени пакета: <software>
2. Если пакет содержит компилируемую часть, то все компилируемые файлы должны входить в пакет <software>-bin
3. Если пакет содержит набор данных отдельно от программной части - изображения, шрифты, база

- данных, то все компоненты должны входить в пакет `<software>-data`
4. Если пакет содержит набор типовых настроек, такие настройки включаются в базовый пакет `<software>`, если вариантов настройки пакета несколько, создаются соответствующие конфигурационные пакеты вида: `<software>-config-<kind>`, где *kind* - разновидность предоставляемых настроек.
 5. Документация на пакет ПО упаковывается в пакет `<software>-doc`
 6. При наличии SDK, такие файлы должны помещаться в пакет `<software>-dev`
 7. Если пакет имеет изменяемую структуру БД (миграции), то такие миграции должны быть включены в базовый пакет `<software>`
 8. Версия пакета формируется автоматически из последнего тега + порядкового номера коммита относительно тега. ХЭШ коммита попадает в сведения о сборке пакета.
 9. Список изменений проекта формируется автоматически из строк фиксации изменений кода в git, при этом учитываются только строки начинающиеся с символов `+/-/*`
 10. Правила формирования пакетов описываются DevOps инженером совместно с руководителем проекта или тимлидом.

Формирование Docker контейнера

Docker файл формирует образ программного обеспечения как результат сборки debian пакета. Docker файл должен предоставлять simple контейнер, из одного или нескольких образов ФС:

- Базовая ОС
- Необходимые зависимости
- Образ с распакованным ПО
- Сценарии при запуске контейнера (преинит)

Итоговый контейнер должен соответствовать следующим принципам:

1. **Монолитность** Отсутствие сборки ПО при установке контейнера
2. **Стабильность** Отсутствие скачивания дополнительных компонент из сети интернет
3. **Повторяемость** Возможность многократно развернуть контейнер в средах отличных от Docker
4. **Атомарность** Иметь отдельные образы контейнеров для нужд БД и рабочей нагрузки
5. **Масштабируемость** Иметь поддержку горизонтального масштабирования (NLB) с общей БД, разделяемой ФС и конфигурацией
6. **Управляемость** Контейнер должен содержать базовый набор Unix Shell утилит, на уровне не ниже Busybox 1.7

Репозиторий образов

Для целей поддержки инфраструктуры Kubernetes для программного обеспечения выделяется общий разделяемый репозиторий корпоративного ПО. Перед передачей ПО в ГосТех или иные сторонние площадки оркестрации и управления ПО **должно** пройти все необходимые процедуры тестирования и приемки в среде **Тестовой** и **Опытной** эксплуатации.

🕒 Версия #2

★ Баталин Иван создал 2 апреля 2024 02:55:22

✍ Баталин Иван обновил 2 апреля 2024 03:10:34