

Сборочный конвейер для реализации CI/CD

Мотивация

На протяжении всего курса для CI/CD мы использовали связку Travis-CI и Bintray. Такая идеология мне нравится и я считаю ее отличной, но довольно часто приходилось сталкиваться с трудностями на стороне Travis-CI. Задача моего проекта — создание своего маленького аналога Travis-CI.

На Travis-CI только дистрибутивы Ubuntu, причем не самые свежие (последняя версия — 18.04), многие пакеты устарели еще три года назад, а обновление и добавление репозитория на Ubuntu лично для меня является болью. К примеру, GTest в репозиториях содержит баг, исправленный три года назад, и из-за него происходит бесконечная рекурсия при обращении к `std::filesystem::path`. К слову, использование `std::filesystem` тоже нетривиально, т.к. доступен только gcc-8 (актуальная версия 10), в котором необходимо вручную линковаться с `stdc++fs`.

Подход CI/CD мне бы хотелось использовать и дальше не только на работе, но и для своих pet-проектов, и иметь на сборочной ферме родное и близкое для меня окружение, позволяющее использовать последние версии компиляторов, библиотеки и прочих тулзов. Таким окружением для меня является Arch Linux. Разумеется, проект подразумевает возможность добавления других окружений (работает все через Docker).

Техническое задание

Опишем основные требования к проекту.

- Конечный продукт представляет собой два исполняемых файла: `ci-cd-server` и `ci-cd-client`. Клиент должен передать серверу параметры, переданные в него в качестве аргументов командной строки: репозиторий на GitHub, ветка репозитория, а также json-файла с приватными переменными окружения, которые нельзя выносить в репозиторий (например, токен Bintray). Разумеется, клиент принимает еще и адрес и порт сервера для связи с ним.

- `ci-cd-server` должен найти в указанном репозитории и ветке на GitHub конфигурационный файл `.build-config.json`. Из файла сервер считывает информацию об используемом дистрибутиве, действиях перед, во время и после сборки. Если файл некорректен или его не существует — пользователь получает сообщение об ошибке.
- Сервер не должен блокироваться при выполнении задачи от одного пользователя, работа должна осуществляться асинхронно. При этом пользователь должен в режиме онлайн видеть логи сборки: запускаемые команды и их вывод.

Текущий результат

На данный момент выполнены все пункты из технического задания. Это позволяет вручную запустить билд на нужной ветке нужного репозитория и автоматически отправить артефакты сборки на Bintray, что и было главной целью. Пример вывода, который получает клиент, можно посмотреть в приложении в конце этого документа.

Будущее развитие

Проект может быть улучшен множеством способов, некоторые правки абсолютно необходимы. Перечислим часть из них:

- В данный момент билды никак не сохраняются: пользователь получает логи сборки, артефакты улетают на Bintray, и на этом все. Необходимо реализовать регистрацию пользователей и сохранение билдов. Вместе с этим добавить нумерацию билдов, решив проблему версионирования
- После добавления регистрации пользователей нужно добавить возможность отслеживать конкретный репозиторий. Это базовая вещь, которую должен уметь делать конвейер, но пока она не реализована.
- Приватные токены передаются по сети в открытом виде, в полноценном продукте это недопустимо. Вместе с добавлением авторизации нужно решить и эту проблему
- Необходимо добавить возможность работы не только с GitHub, но и с другими хостингами исходного кода.

- Сериализация данных для передачи от клиента серверу происходит вручную и слабо соответствует принципу DRY. Нужно найти способ решения этой проблемы.
- На данном этапе проект достаточно простой, и реализация уместилась в несколько компактных классов без применения какой-то мудреной архитектуры. В будущем, скорее всего, придется пересмотреть структуру исходного кода для упрощения расширения. Для этого сначала необходимо понять, как вообще может захотеться расширять код.

Приложение

Пример вывода, который получает клиент:

```
1 $ ./ci-cd-client --ip 127.0.0.1 --port 9000 --repo kovdan01/test-repo --branch master --env tokens.  
  json  
2  
3 SET PUBLIC ENV VARIABLES  
4  
5 > export CI_CD_BRANCH=master  
6  
7 INIT DOCKER  
8  
9 > pacman -Sy --noconfirm  
10 :: Synchronizing package databases...  
11 downloading core.db...  
12 downloading extra.db...  
13 downloading community.db...  
14 > pacman -S git --noconfirm  
15 resolving dependencies...  
16 looking for conflicting packages...  
17  
18 Packages (4) perl-error-0.17029-2  perl-mailtools-2.21-4  perl-timedate-2.33-2  git-2.27.0-1  
19  
20 Total Download Size:    6.68 MiB  
21 Total Installed Size:  41.49 MiB  
22  
23 :: Proceed with installation? [Y/n]  
24 :: Retrieving packages...  
25 downloading perl-error-0.17029-2-any.pkg.tar.zst...  
26 downloading perl-timedate-2.33-2-any.pkg.tar.zst...  
27 downloading perl-mailtools-2.21-4-any.pkg.tar.zst...  
28 downloading git-2.27.0-1-x86_64.pkg.tar.zst...  
29 checking keyring...  
30 checking package integrity...  
31 loading package files...  
32 checking for file conflicts...  
33 :: Processing package changes...  
34 installing perl-error...  
35 installing perl-timedate...  
36 installing perl-mailtools...  
37 installing git...  
38 Optional dependencies for git  
39  tk: gitk and git gui  
40  perl-libwww: git svn  
41  perl-term-readkey: git svn and interactive.singlekey setting  
42  perl-mime-tools: git send-email  
43  perl-net-smtp-ssl: git send-email TLS support  
44  perl-authen-sasl: git send-email TLS support  
45  perl-mediawiki-api: git mediawiki support  
46  perl-datetime-format-iso8601: git mediawiki support  
47  perl-lwp-protocol-https: git mediawiki https support  
48  perl-cgi: gitweb (web interface) support  
49  python: git svn & git p4  
50  subversion: git svn  
51  org.freedesktop.secrets: keyring credential helper  
52  libsecret: libsecret credential helper [installed]
```

Проектная работа на курсе Otus «Разработчик C++». Ковалев Даниил

```
53 :: Running post-transaction hooks...
54 (1/4) Creating system user accounts...
55 Creating group git with gid 977.
56 Creating user git (git daemon user) with uid 977 and gid 977.
57 (2/4) Reloading system manager configuration...
58   Skipped: Current root is not booted.
59 (3/4) Arming ConditionNeedsUpdate...
60 (4/4) Warn about old perl modules
61 WARNING: '/usr/lib/perl5/5.32' contains data from at least 3 packages which will NOT be used by the
    installed perl interpreter.
62   -> Run the following command to get a list of affected packages: pacman -Qqo '/usr/lib/perl5/5.32'
63
64 PRE-BUILD ACTIONS
65
66 > pacman -S gcc make cmake --noconfirm
67 resolving dependencies...
68 looking for conflicting packages...
69
70 Packages (19) binutils-2.34-5  elfutils-0.180-1  gc-8.0.4-4  guile-2.2.6-2  icu-67.1-1  jsoncpp-1.9.2
    -1  libelf-0.180-1  libmicrohttpd-0.9.70-3  libmpc-1.1.0-2  libnsl-1.2.0-2  libtool-2.4.6+42+
    gb88ceb5-13  libuv-1.38.0-1  libxml2-2.9.10-2  rhash-1.3.9-2  shared-mime-info-2.0+1+g6bf9e4f-2
    texinfo-6.7-3  cmake-3.17.3-1  gcc-10.1.0-2  make-4.3-3
71
72 Total Download Size:    68.33 MiB
73 Total Installed Size:  336.47 MiB
74 Net Upgrade Size:      333.66 MiB
75
76 :: Proceed with installation? [Y/n]
77 :: Retrieving packages...
78 downloading libmicrohttpd-0.9.70-3-x86_64.pkg.tar.zst...
79 downloading libelf-0.180-1-x86_64.pkg.tar.zst...
80 downloading elfutils-0.180-1-x86_64.pkg.tar.zst...
81 downloading binutils-2.34-5-x86_64.pkg.tar.zst...
82 downloading libmpc-1.1.0-2-x86_64.pkg.tar.xz...
83 downloading gcc-10.1.0-2-x86_64.pkg.tar.zst...
84 downloading libtool-2.4.6+42+gb88ceb5-13-x86_64.pkg.tar.zst...
85 downloading texinfo-6.7-3-x86_64.pkg.tar.zst...
86 downloading make-4.3-3-x86_64.pkg.tar.zst...
87 downloading icu-67.1-1-x86_64.pkg.tar.zst...
88 downloading libnsl-1.2.0-2-x86_64.pkg.tar.xz...
89 downloading gc-8.0.4-4-x86_64.pkg.tar.zst...
90 downloading guile-2.2.6-2-x86_64.pkg.tar.zst...
91 downloading libxml2-2.9.10-2-x86_64.pkg.tar.zst...
92 downloading shared-mime-info-2.0+1+g6bf9e4f-2-x86_64.pkg.tar.zst...
93 downloading jsoncpp-1.9.2-1-x86_64.pkg.tar.zst...
94 downloading libuv-1.38.0-1-x86_64.pkg.tar.zst...
95 downloading rhash-1.3.9-2-x86_64.pkg.tar.zst...
96 downloading cmake-3.17.3-1-x86_64.pkg.tar.zst...
97 checking keyring...
98 checking package integrity...
99 loading package files...
100 checking for file conflicts...
101 :: Processing package changes...
102 installing libmicrohttpd...
103 upgrading libelf...
104 installing elfutils...
105 installing binutils...
106 installing libmpc...
107 installing gcc...
108 Optional dependencies for gcc
109   lib32-gcc-libs: for generating code for 32-bit ABI
110 installing libtool...
111 installing texinfo...
112 installing gc...
113 installing guile...
114 installing make...
115 installing icu...
116 installing libxml2...
117 installing shared-mime-info...
118 installing jsoncpp...
119 Optional dependencies for jsoncpp
120   jsoncpp-doc: documentation
121 installing libnsl...
122 installing libuv...
123 installing rhash...
124 installing cmake...
125 Optional dependencies for cmake
126   qt5-base: cmake-gui
127 :: Running post-transaction hooks...
128 (1/2) Arming ConditionNeedsUpdate...
129 (2/2) Updating the MIME type database...
```

Проектная работа на курсе Otus «Разработчик C++». Ковалев Даниил

```
130 |
131 | GETTING SOURCES
132 |
133 | > git clone --single-branch --branch master https://github.com/kovdan01/test-repo.git repo
134 | > cd repo
135 |
136 | BUILD ACTIONS
137 |
138 | > mkdir build
139 | > cd build
140 | > cmake ..
141 | -- The CXX compiler identification is GNU 10.1.0
142 | -- Check for working CXX compiler: /usr/sbin/c++
143 | -- Check for working CXX compiler: /usr/sbin/c++ - works
144 | -- Detecting CXX compiler ABI info
145 | -- Detecting CXX compiler ABI info - done
146 | -- Detecting CXX compile features
147 | -- Detecting CXX compile features - done
148 | -- Configuring done
149 | -- Generating done
150 | -- Build files have been written to: /repo/build
151 | > cmake --build .
152 | Scanning dependencies of target test-repo
153 | [ 50%] Building CXX object CMakeFiles/test-repo.dir/main.cpp.o
154 | [100%] Linking CXX executable test-repo
155 | [100%] Built target test-repo
156 | > cmake --build . --target package
157 | [100%] Built target test-repo
158 | Run CPack packaging tool...
159 | CPack: Create package using DEB
160 | CPack: Install projects
161 | CPack: - Run preinstall target for: test-repo
162 | CPack: - Install project: test-repo []
163 | CPack: Create package
164 | -- CPackDeb: Can not find dpkg in your path, default to i386.
165 | -- CPACK_DEBIAN_PACKAGE_DEPENDS not set, the package will have no dependencies.
166 | CPack: - package: /repo/build/test-repo-0.0.1-Linux.deb generated.
167 |
168 | POST-BUILD ACTIONS
169 |
170 | > curl -T test-repo-0.0.1-Linux.deb -ukovdan01:$BINTRAY_API_KEY "https://api.bintray.com/content/
    kovdan01/test-repo/$CI_CD_BRANCH/2/test-repo-0.0.2-Linux.deb;deb_distribution=trusty;
    deb_component=main;deb_architecture=amd64;publish=1"
171 | {"message": "success"}
```