

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

«Digit Смарт-БД»

Сведения о хранении и компиляции

Листов 7

СОДЕРЖАНИЕ

1 ХРАНЕНИЕ ИСХОДНОГО КОДА	4
2 ОСОБЕННОСТИ РЕПОЗИТОРИЯ.....	5
3 КОМПИЛЯЦИЯ ИСХОДНОГО КОДА	6
4 СБОРКА И РАЗВЕРТЫВАНИЕ С ИСПОЛЬЗОВАНИЕМ DOCKER	7
1 Процесс сборки	7
Этапы пайплайна	7
Развертывание.....	7
2 Артефакт поставки.....	7

СПИСОК СОКРАЩЕНИЙ

Таблица 1 – Список терминов и сокращений

СОКРАЩЕНИЕ	НАИМЕНОВАНИЕ
GIT	Распределенная система управления версиями, которая отслеживает и фиксирует изменения в файлах (чаще всего речь идет об исходном коде программ, однако систему можно использовать для файлов любого назначения).
GITLAB	Инструмент хранения и версионирования исходного кода с использованием системы контроля версий git. Кроме того, инструмент под названием gitlab ci, интегрированный в gitlab, используется для автоматизации выполнения непрерывного релизного цикла продуктов, исходный код которых разрабатывается в gitlab.
GITLAB RUNNER	Корпоративная сеть передачи данных – телекоммуникационная сеть, объединяющая в единое информационное пространство все краевые структурные подразделения органов власти
GITLAB CI	Инструмент автоматизации выполнения непрерывного релизного цикла продуктов, исходный код которых разрабатывается в gitlab
PIPELINE	Цепочка задач, организованных в последовательные этапы, ветвление на параллельно запускаемые этапы также возможно. Цепочка конфигурируется из входных параметров, подготовленных ci/cd системой на основании определяемых начальных условий, например, ветка/тег git-репозитория
YAML	Формат данных, близкий к языкам разметки, но ориентированный на удобство ввода-вывода типичных структур данных многих языков программирования. Формат удобен для редактирования, поэтому активно используется как один из основных конфигурационных форматов.
РЕПОЗИТОРИЙ	Централизованное хранилище исходного кода в git, Проект в gitlab

1 ХРАНЕНИЕ ИСХОДНОГО КОДА

Для разработки программного обеспечения «Digit Смарт-БД» используется система контроля версий Git.

Хранение исходного кода осуществляется в локально развернутом репозитории GitLab.

GitLab используется как единая платформа для:

- хранения исходного кода;
- управления версиями;
- совместной разработки;
- отслеживания изменений;
- ведения истории разработки.

Взаимодействие разработчиков с системой осуществляется с использованием:

- консольных инструментов Git;
- веб-интерфейса GitLab.

Для автоматизации процессов разработки применяется система непрерывной интеграции GitLab CI/CD, обеспечивающая:

- автоматическую сборку программного обеспечения;
- проверку исходного кода;
- формирование артефактов;
- публикацию результатов сборки.

Запуск процессов CI/CD осуществляется по событиям:

- фиксация изменений (commit);
- слияние веток;
- создание тегов релизов.

Конфигурация процессов CI/CD задается в файле `.gitlab-ci.yml`.

2 ОСОБЕННОСТИ РЕПОЗИТОРИЯ

В рамках проекта «Digit Смарт-БД» используется централизованная структура репозитория в GitLab.

GitLab обеспечивает:

- управление репозиториями исходного кода;
- управление доступом пользователей;
- разграничение прав доступа;
- отслеживание изменений и активности;
- интеграцию с CI/CD.

Используется ролевая модель доступа:

- **Owner** — владелец проекта;
- **Maintainer** — ответственный за проект;
- **Developer** — разработчик;
- **Reporter** — пользователь с правом просмотра;
- **Guest** — ограниченный доступ.

Основные сущности:

Проект — содержит:

- исходный код;
- конфигурацию CI/CD;
- историю изменений;
- задачи и релизы.

Группа — используется для объединения проектов и управления доступом.

3 КОМПИЛЯЦИЯ ИСХОДНОГО КОДА

Сборка программного обеспечения «Digit Смарт-БД» выполняется автоматически с использованием GitLab CI/CD.

Процесс компиляции включает:

1. Получение исходного кода из репозитория;
2. Выполнение сборки серверной части (Java, Spring Boot);
3. Сборка клиентской части (Angular);
4. Формирование итоговых артефактов.

Конфигурация сборки определяется в файле `.gitlab-ci.yml`.

Типовой пайплайн включает этапы:

- **build** — компиляция исходного кода и сборка артефактов;
- **test** — выполнение тестов и проверок;
- **package** — формирование итогового дистрибутива.

Результатом сборки являются:

- исполняемый файл серверной части (JAR);
- статические файлы клиентского интерфейса (frontend build);
- вспомогательные конфигурационные файлы.

4 СБОРКА И РАЗВЕРТЫВАНИЕ С ИСПОЛЬЗОВАНИЕМ DOCKER

В проекте «Digit Смарт-БД» используется контейнеризация приложений.

Сборка и развертывание выполняются с использованием Docker.

1 Процесс сборки

В рамках CI/CD выполняются следующие действия:

- сборка Docker-образа на основе Dockerfile;
- включение серверной части и клиентского интерфейса в образ;
- присвоение тегов образу;
- публикация образа в реестр контейнеров.

Этапы пайплайна

- **test** — запуск тестов;
- **build-image** — сборка Docker-образа;
- **publish** — публикация образа в реестр;
- **deploy** — развертывание на целевом сервере.

Развертывание

Развертывание осуществляется:

- загрузкой образа из реестра;
- остановкой предыдущей версии сервиса;
- запуском контейнера с новой версией.

Управление контейнерами выполняется с использованием Docker или docker-compose.

2 Артефакт поставки

В качестве артефакта поставки используется:

- Docker-образ приложения;

или

- дистрибутив, включающий серверную часть и клиентский интерфейс.