

Содержание

ВВЕДЕНИЕ	2
1 Основная часть	3
1.1 МОБД по ЭКБ КП	3
1.2 Архитектура хранилища МОБД по ЭКБ КП	4
1.3 Детали реализации	4
1.3.1 Версионирование и поддержка CUBA	4
1.3.2 Sonatype Nexus	5
1.4 Набор клиентских средств для выкладки ПАК	5
1.4.1 Компоненты Docker, используемые в выкладке	6
1.5 Процесс выкладки программного окружения	6
1.5.1 Окружение, необходимое для сборки артефактов	8
1.5.2 Сборка подсистем отраслевой БД по ЭКБ КП	9
ЗАКЛЮЧЕНИЕ	10
Список литературы	11

ВВЕДЕНИЕ

Модернизированная отраслевая база данных электронной компонентной базы космического применения (МОБД ЭКБ КП) позволяет разработчикам радиоэлектронной аппаратуры получить информацию о результатах испытаний компонентов ЭКБ на радиационную стойкость. База связана с отраслевой информационно-справочной системой Госкорпорации «Роскосмос» по стойкости ЭКБ. Программный комплекс, предназначенный для обеспечения доступа разработчиков к модернизируемой отраслевой БД включает обширный комплекс ресурсов, которые должны взаимодействовать в рабочем окружении.

Цель производственной, эксплуатационной практики – выкладка программного обеспечения ПАК МОБД ЭКБ в рабочем окружении. Для достижения поставленной цели необходимо выполнить следующие задачи:

- изучить хранилище данных МОБД ЭКБ КП;
- изучить программные средства, с помощью которых реализован программный комплекс;
- обеспечить набор клиентских средств, позволяющий произвести выкладку всех элементов программного комплекса в рабочем окружении;
- произвести упаковку зависимостей программного комплекса в локальном рабочем окружении.

1. ОСНОВНАЯ ЧАСТЬ

1.1 МОБД по ЭКБ КП

Модернизируемая отраслевая БД по ЭКБ КП состоит из следующих модулей(подсистем):

- хранилища данных;
- подсистемы сбора и учета данных;
- подсистемы формирования выходных отчетов;
- подсистемы информационной безопасности;
- подсистемы администрирования.

Хранилище данных предназначено для накопления и хранения данных элементов ЭКБ. Хранилище реализовано в виде базы данных. Базой данных является представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ). [1]

Подсистема сбора и учета данных предназначена для сбора и загрузки в Модернизированную отраслевую БД по ЭКБ КП, а также выгрузки из нее информации, необходимой для автоматизации процесса оценки и выбора элементов ЭКБ КП.

Подсистема информационной безопасности предназначена для более обширного спектра задач. Пользователь имеет возможность аутентификации с помощью программных средств. Также подсистема информационной безопасности обеспечивает защиту ПАК от несанкционированного доступа к БД по ЭКБ КП.

Подсистема администрирования обеспечивает контроль за действиями пользователей в системе, создание и удаление пользователей, возможность разграничения доступа пользователей к данным.

1.2 АРХИТЕКТУРА ХРАНИЛИЩА МОБД ПО ЭКБ КП

Хранилище данных МОБД по ЭКБ КП реализовано посредством технологии «клиент-сервер».

1.3 ДЕТАЛИ РЕАЛИЗАЦИИ

Подсистемы сбора и учета данных, формирования выходных отчетов, информационной безопасности и администрирования реализованы на языке программирования Java, используя платформу для разработки бизнес-приложений CUBA. [2] Платформа имеет возможность нативной поддержки PostgreSQL, [3] которая используется в качестве системы управления базами данных для базы данных элементов ЭКБ.

1.3.1 ВЕРСИОНИРОВАНИЕ И ПОДДЕРЖКА CUBA

Нумерация стабильных версий CUBA Platform формируется в соответствии с традиционным семантическим версионированием [4]:

major.minor.maintenance,

где:

- *maintenance* – обновление устранения неполадок. Обеспечивает обратную совместимость. Включает незначительные дополнительные возможности или улучшения, исправление дефектов, незначительные обновления, критические обновления для производительности и безопасности. Такие обновления не несут существенных изменений.
- *minor* – обновление, в основном совместимое с предыдущими версиями, однако может привносить существенные изменения на уровне основных возможностей. Предназначение *minor*-релиза – введение новых возможностей при быстром процессе обновления.

- *major* – основное обновление. Включает в себя несовместимые изменения базовой архитектуры, функциональных возможностей, изменения на уровне программного интерфейса приложения, лежащего в основе библиотек и их версий. Для основных обновлений обратная совместимость необязательна.

ПАК МОБД по ЭКБ КП – проект с длительным циклом обновления. Поэтому подсистемы реализованы с использованием версии, для которой осуществляется только корпоративная поддержка. Для того, чтобы компоненты приложения могли функционировать в системе, необходим приватный репозиторий артефактов.

1.3.2 SONATYPE NEXUS

Sonatype Nexus – интегрированная платформа, с помощью которой разработчики могут хранить и управлять локальными зависимостями Java (Maven). Выбор платформы обусловлен тем, что локальные артефакты, хранимые в репозитории недоступны из внешних репозиториях.

1.4 НАБОР КЛИЕНТСКИХ СРЕДСТВ ДЛЯ ВЫКЛАДКИ ПАК

Конфигурация среды разработки зависит от платформы, на которой происходит эта самая разработка. Поэтому решено использовать механизм контейнеризации, чтобы абстрагироваться от имеющегося окружения и работать с неизменной и защищенной средой внутри контейнера.

Docker [5] – это платформа контейнеризации с открытым исходным кодом, с помощью которой можно автоматизировать создание приложений, их доставку и управление.

1.4.1 КОМПОНЕНТЫ DOCKER, ИСПОЛЬЗУЕМЫЕ В ВЫКЛАДКЕ

В таблице 1.1 даны определения основных компонентов Docker, используемых при выкладке.

Таблица 1.1 – Компоненты Docker, используемые в выкладке

Имя компонента	Описание	Применение
Dockerfile	Текстовый файл с последовательно расположенными инструкциями для создания образа Docker	Развертывание окружения, необходимого для сборки артефактов
image	Неизменяемый файл (образ), из которого можно неограниченное количество раз развернуть контейнер	Использованы образы окружения сборки, репозитория Nexus
docker-compose.yml	Определение служб для централизованного запуска при сборке многоконтейнерного приложения Docker	Централизованный запуск служб Nexus, СУБД Postgres и окружения сборки
docker volumes	Тома для постоянного хранения информации	Хранение изменяемых данных после остановки службы

1.5 ПРОЦЕСС ВЫКЛАДКИ ПРОГРАММНОГО ОКРУЖЕНИЯ

На листинге 1.1 приведено определение служб в файле docker-compose для централизованного запуска при сборке многоконтейнерного приложения Docker.

Листинг 1.1 – файл docker-compose.yml

```
1 version: '3.7'
2
3 services:
```

```

4  nexus:
5      build: nexus
6      container_name: nexus
7      ports:
8          - '8081:8081'
9      volumes:
10         - ./nexus/data:/nexus-data
11 database:
12     image: postgres:10-alpine
13     container_name: database
14     ports:
15         - '5432:5432'
16     volumes:
17         - ./database/data:/var/lib/postgresql/data
18     environment:
19         - POSTGRES_PASSWORD=postgres
20 builder:
21     build: builder
22     container_name: builder
23     network_mode: host
24     volumes:
25         - ./sources:/home/gradle
26     depends_on:
27         - nexus
28         - database

```

Корневой ключ в этом файле – `services`. Под этим ключом определяются службы, которые требуется развернуть и запустить. В данном случае в файле `docker-compose.yml` определено несколько служб, как показано в таблице 1.2.

Таблица 1.2 – Назначение служб

Имя службы	Описание
nexus	Организация локального хранилища пакетов и артефактов CUBA, а также управление запросами и перенаправление их на специальные датацентры РКС
database	Развертывание реляционной СУБД PostgreSQL
builder	Окружение, необходимое для корректной сборки артефактов системы

1.5.1 ОКРУЖЕНИЕ, НЕОБХОДИМОЕ ДЛЯ СБОРКИ АРТЕФАКТОВ

На листинге 1.2 приведено содержание файла Dockerfile для развертывания окружения, необходимого для сборки артефактов.

Листинг 1.2 – Dockerfile для для службы builder

```
1 FROM eclipse-temurin:8-jdk-jammy
2
3 CMD ["bash"]
4
5 ENV GRADLE_HOME /opt/gradle
6
7 RUN set -o errexit -o nounset \
8     && echo "Adding gradle user and group" \
9     && groupadd --system --gid 1000 gradle \
10    && useradd --system --gid gradle --uid 1000 ... \
11    && mkdir /home/gradle/.gradle \
12    && chown --recursive gradle:gradle /home/gradle \
13    \
14    && echo "Symlinking root Gradle cache to gradle Gradle cache" \
15    && ln --symbolic /home/gradle/.gradle /root/.gradle
16
17 VOLUME /home/gradle/.gradle
18
19 WORKDIR /home/gradle
20
21 RUN set -o errexit -o nounset \
22     && apt-get update \
23     && apt-get install --yes --no-install-recommends \
24         unzip \
25         wget \
26         \
27         bzip \
28         git \
29         git-lfs \
30         mercurial \
31         openssh-client \
32         subversion \
33     && rm --recursive --force /var/lib/apt/lists/* \
34     \
35     && echo "Testing VCSes" \
36     && which bzip \
37     && which git \
38     && which git-lfs \
```



```

39  && which hg \
40  && which svn
41
42  ENV GRADLE_VERSION 4.3.1
43  ARG GRADLE_DOWNLOAD_SHA256=15ebe098ce0392a2d06...
44  RUN set -o errexit -o nounset \
45  && echo "Downloading Gradle" \
46  && wget --no-verbose --output-document=gradle.zip ... \
47  \
48  && echo "Checking download hash" \
49  && echo "${GRADLE_DOWNLOAD_SHA256} *gradle.zip" | sha256sum --check - \
50  \
51  && echo "Installing Gradle" \
52  && unzip gradle.zip \
53  && rm gradle.zip \
54  && mv "gradle-${GRADLE_VERSION}" "${GRADLE_HOME}/" \
55  && ln --symbolic "${GRADLE_HOME}/bin/gradle" /usr/bin/gradle \
56  \
57  && echo "Testing Gradle installation" \
58  && gradle --version

```

1.5.2 СБОРКА ПОДСИСТЕМ ОТРАСЛЕВОЙ БД по ЭКБ КП

После успешного запуска всех необходимых служб с помощью задач в системе автоматической сборки «Gradle» осуществляется сборка подсистем в строгом порядке, поскольку на каждом этапе определенная подсистема отправляет в удаленный репозиторий Nexus артефакты, требуемые для дальнейшей выкладки.

Основной модуль запускается с помощью «Apache Tomcat» – программы, представляющая собой сервер, который занимается системной поддержкой сервлетов и обеспечивает их жизненный цикл в соответствии с правилами, определёнными в спецификациях. [6]

Выкладка программного комплекса на данном этапе завершена и программное обеспечение ПАК МОБД ЭКБ доступно по адресу, указанному в файле docker-compose для централизованного запуска служб приложения (листинг 1.1).

ЗАКЛЮЧЕНИЕ

Модернизируемая отраслевая БД по ЭКБ КП обеспечивает перевод бумажного документооборота в электронный вид. Поскольку стандарты документов меняются, а электронная компонентная база космического применения стремительно развивается, задачи модернизации, развития и поддержки программного обеспечения ЭКБ КП остаются актуальными. Автоматизация выкладки элементов программного комплекса значительно упростит процессы, связанные с перечисленными задачами.

Цель производственной, эксплуатационной практики достигнута – произведена выкладка программного обеспечения ПАК МОБД ЭКБ в рабочем окружении. Поставленные во введении задачи были достигнуты.

1. Изучено хранилище данных МОБД ЭКБ КП, выделены основные компоненты системы.
2. Изучены программные средства, с помощью которых реализован программный комплекс.
3. Обеспечен набор клиентских средств, позволивший произвести выкладку элементов программного комплекса в рабочем окружении.
4. Произведена упаковка зависимостей программного комплекса в локальном рабочем окружении.

СПИСОК ЛИТЕРАТУРЫ

1. Гражданский кодекс РФ, ст. 1260.
2. CUBA Platform. Developer's Manual. — URL: <https://doc.cuba-platform.com/manual-latest/> (дата обр. 14.07.2022).
3. PostgreSQL: The World's Most Advanced Open Source Relational Database. — URL: <https://www.postgresql.org/> (дата обр. 14.07.2022).
4. Навигация по стабильным версиям. — URL: <https://www.jmix.ru/cuba-platform/framework/versioning/> (дата обр. 14.07.2022).
5. Docker docs. — URL: <https://docs.docker.com/> (дата обр. 14.07.2022).
6. Apache Tomcat®. — URL: <https://tomcat.apache.org/> (дата обр. 14.07.2022).