ПРОГРАММНОЕ ИЗДЕЛИЕ

«AXIOM RUNTIME CONTAINER PRO»

Руководство администратора

Листов 6

2022

АННОТАЦИЯ

Настоящий документ содержит руководство администратора изделия «Axiom Runtime Container Pro».

1. Введение

Axiom Runtime Container Pro включает в свой состав инструменты (утилиты, программную среду) ОС БЕЛЛСОФТ Linux и «Axiom JDK Pro» и поставляется в виде готового контейнера.

1.1. Docker

Docker — это платформа контейнеризации для создания, совместного использования и запуска приложений.

Docker позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть запущен на любой системе, а также предоставляет набор команд для управления этими контейнерами. Приложения Docker можно использовать в любой системе: ноутбуке, настольных системах, серверах или в облаке. Контейнер — это стандартная единица программного обеспечения, которая упаковывает код и все его зависимости, чтобы приложение быстро и надежно запускалось в вычислительной среде. Docker предоставляет набор инструментов для создания и совместного использования образов контейнеров, а также запуска контейнеров в любом масштабе.

Для получения дополнительной информации посетите сайт Docker.

1.2. Axiom JDK Pro

Среда разработки и исполнения JAVA™ Axiom JDK Pro — это Java SE дистрибутив с экспертной технической поддержкой, прошедший проверку на соответствие спецификациям Java SE. Axiom JDK Pro — отечественная альтернатива инструментальным средствам разработки приложений на Java™ для коммерческих и государственных компаний.

Axiom JDK Pro в контейнере Axiom Runtime Container Pro поддерживает архитектуру x86_64 и AArch64.

Axiom создается, тестируется, поддерживается и предоставляется БЕЛЛСОФТ.

1.3. БЕЛЛСОФТ Linux

БЕЛЛСОФТ Linux – полнофункциональная операционная система, оптимизированная для Java. Данную ОС характеризуют следующие отличительные свойства.

Безопасность

Усовершенствования БЕЛЛСОФТ Linux ОС включают сетевые функции, настраиваемые параметры сборки, усиление защиты ядра, изоляцию процессов и многое другое.

Производительность

БЕЛЛСОФТ Linux оптимизирован для предоставления надежного и быстрого стека для запуска приложений для различных рабочих нагрузок. Он включает в себя функции производительности, такие как оптимизация ядра, управление памятью и оптимизация реализации libc, включая версию musl с переработанными API.

Функциональность

БЕЛЛСОФТ Linux оптимизирован для развертывания за счет использования поддержки сжатия модулей ядра для уменьшения размера пакетов в репозитории и пакетов, установленных в системе.

Дополнительные модули и пакеты прошивки разделены на легкие пакеты и могут быть установлены в зависимости от потребностей и целевой платформы.

2. Образы Docker

БЕЛЛСОФТ предоставляет ряд образов Docker. Они размещены в библиотеке/репозитории образов контейнеров, доступной по адресу, который БЕЛЛСОФТ сообщает каждому клиенту.

Этот репозиторий содержит образы Docker с Axiom JDK Pro и доступен для следующих архитектур:

- x86_64 (он же amd64)
- AArch64

Репозиторий содержит ряд тегов. Каждый тег указывает на образ Docker.

2.1. Авторизация

Как правило, загрузка общедоступных образов не требует авторизации.

Если БЕЛЛСОФТ предоставляет ссылку на частное пространство для доступа к предварительной версии программного обеспечения или другой специальной версии, авторизуйтесь, прежде чем извлекать образ. Для этого выполните следующую команду, а затем введите имя пользователя и пароль.

```
docker login <imagerepo>
```

Где `<imagerepo>` это URL-адрес репозитория.

3. Создание контейнера и запуск

Чтобы создать контейнер из образа с тегом `<image_tag>` из репозитория `<repository>`, выполните одну из следующих команд.

Примечание: `<imagerepo>` это URL-адрес репозитория.

- Скачайте образ используя команду docker pull:

```
docker image pull <imagerepo>/<repository>:<image_tag>
```

и запустите контейнер с помощью команды docker run:

```
docker container run --rm -it <imagerepo>/<repository>:<image_tag>
```

- Вы также можете объединить две предыдущие операции в одну команду `docker run`:

```
docker run --rm -it <imagerepo>/<repository>:<image_tag>
```

Если часть `:<image_tag>` опущена, она заменяется на `:latest`, то есть команда работает с тегом `latest`.

Пример запуска контейнера Axiom JDK Pro 11:

```
docker run --rm -it <imagerepo>/axiom-runtime-container-pro:jdk-11musl
```

Для запуска какого-либо приложения вы можете создать Dockerfile на основе образа `axiom-runtime-container-pro` или смонтировать том с вашим кодом/приложением, например:

```
docker run -it --rm -v /home/user/project/:/data
<imagerepo>/axiomruntime-container-pro:latest java -jar
/data/MyApp.jar
```

Описание доступного образа представлено ниже.

4. Образ axiom-runtime-container-pro

Репозиторий содержит образы, поставляемые с программным обеспечением Axiom JDK Pro и JRE.

Для версий Java 8, 11 и 17 доступны следующие теги в зависимости от версии musl или glibc.

Образы с последней версией Axiom JRE для определенной версии Java, где <x> - версия.

- jre-<x>-musl
- jre-<x>-glibc

Образы с последней версией Axiom JDK Pro для определенной версии Java, где <x> - версия.

- jdk-<x>-musl

- jdk-<x>-glibc

```
 $ docker run --rm <imagerepo>/axiom-runtime-container-pro:jre-11-musl
java -version
 axiomjdk version "11.0.14" 2022-01-18 LTS  AxiomJDK Runtime
Environment (build 11.0.14+9-LTS)
 AxiomJDK 64-Bit Server VM (build 11.0.14+9-LTS, mixed mode)
```