AXIOM JDK

Axiom Trusted Images 25 LTS Руководство пользователя

Axiom Trusted Images | Октябрь 2026

Copyright © 2019-2026 Все права защищены АО "АКСИОМ" (АКСИОМ)

Программное обеспечение АКСИОМ содержит программное обеспечение с открытым исходным кодом. Дополнительная информация о коде сторонних разработчиков доступна на сайте https://axiomjdk.ru/third_party_licenses. Для дополнительной информации о том, как получить копию исходного кода, можно обратиться по адресу info@axiomjdk.ru.

ДАННАЯ ИНФОРМАЦИЯ МОЖЕТ ИЗМЕНЯТЬСЯ БЕЗ ПРЕДВАРИТЕЛЬНОГО УВЕДОМЛЕНИЯ. АКСИОМ ПРЕДОСТАВЛЯЕТ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ "КАК ЕСТЬ" БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, АКСИОМ ПРЯМО ОТКАЗЫВАЕТСЯ ОТ ВСЕХ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ПОДРАЗУМЕВАЕМЫМИ ГАРАНТИЯМИ ТОВАРНОЙ ПРИГОДНОСТИ И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ.

АКСИОМ НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ЗА ЛЮБЫЕ КОСВЕННЫЕ, СЛУЧАЙНЫЕ, СПЕЦИАЛЬНЫЕ, ШТРАФНЫЕ ИЛИ КОСВЕННЫЕ УБЫТКИ, ИЛИ УБЫТКИ ОТ ПОТЕРИ ПРИБЫЛИ, ДОХОДА, ДАННЫХ ИЛИ ИСПОЛЬЗОВАНИЯ ДАННЫХ, ПОНЕСЕННЫЕ ВАМИ ИЛИ ЛЮБОЙ ТРЕТЬЕЙ СТОРОНОЙ, БУДЬ ТО В РЕЗУЛЬТАТЕ ДЕЙСТВИЯ ДОГОВОРА ИЛИ ДЕЛИКТА, ДАЖЕ ЕСЛИ АКСИОМ БЫЛО ПРЕДУПРЕЖДЕНО О ВОЗМОЖНОСТИ ТАКИХ УБЫТКОВ.

Использование любого программного продукта АКСИОМ регулируется соответствующим лицензионным соглашением, которое никоим образом не изменяется условиями данного уведомления. Программные продукты и фирменные наименования: Axiom JDK, Axiom JDK Pro, Axiom Runtime Container Pro, Axiom Linux, Libercat, Libercat Certified и АКСИОМ принадлежат АКСИОМ и их использование допускается только с разрешения правообладателя.

Товарный знак Linux® используется в соответствии с сублицензией от Linux Foundation, эксклюзивного лицензиата Линуса Торвальдса, владельца знака на всемирной основе. Java и OpenJDK являются товарными знаками или зарегистрированными товарными знаками компании Oracle и/или ее аффилированных лиц. Другие торговые марки являются собственностью их соответствующих владельцев и используются только в целях идентификации.

Содержание

1. Введение 5

1.1. Что такое Axiom Trusted Images 25 LTS? 5

1.2. Основные концепции 5

1.2.1. Docker 5

1.2.2. Podman 5

1.3. Axiom JDK Pro 6

1.4. Axiom Linux 6

2. Начало работы с "Axiom Trusted Images" 7

2.1. Системные и программные требования 7

2.2. Создание контейнера и запуск 7

2.3. Образы и работа с ними 8

2.3.1. 25-trusted-base 8

2.3.2. 25-trusted-axiom-runtime-container-pro 8

2.3.3. 25-trusted-axiom-runtime-container-certified 8

2.3.4. 25-trusted-axiom-runtime-container-express 9

2.3.5. 25-trusted-axiom-native-image-kit-container 9

2.3.6. 25-trusted-gcc 9

2.3.7. 25-trusted-go 10

2.3.8. 25-trusted-python 11

2.3.9. 25-trusted-nginx 11

2.3.10. 25-trusted-node 11

1. Введение

Настоящий документ содержит начальные сведения по эксплуатации Axiom Trusted Images 25 LTS.

1.1. Что такое Axiom Trusted Images 25 LTS?

Axiom Trusted Images 25 LTS — это набор готовых контейнеров с образами на базе ОС Axiom Linux. Образы могут включать Axiom JDK Pro, Axiom NIK Pro, интерпретатор языка Python, компиляторы GCC и Go, среду выполнения NodeJS и другие утилиты и библиотеки. См. Образы и работа с ними.

1.2. Основные концепции

1.2.1. Docker

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

Docker позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть развёрнут 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 переносилось из одной вычислительной среды в другую. 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.

1.2.2. Podman

Podman – это инструмент с открытым исходным кодом для поиска, сборки, передачи и запуска приложений. Является утилитой командной строки с аналогичными docker командами, однако не требует дополнительный сервис для работы и может работать без прав доступа root.

Для получения дополнительной информации посетите сайт Podman.

1.3. 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 Trusted Images 25 LTS — 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.

1.4. Axiom Linux

Axiom Linux – это современная полнофункциональная операционная система. Данную ОС характеризуют следующие отличительные свойства.

- Надежная многозадачная многопользовательская ОС для серверных задач и хостинга контейнерных нагрузок, в том числе на базе Java;
- Установка с использованием libc библиотеки musl;
- Осуществляет эффективное управление памятью и предоставляет дополнительные реализации malloc, которые могут быть полезны для различных типов нагрузок;
- Поддерживает различные файловые системы во время установки ОС и для настройки дополнительных хранилищ;
- Предоставляет расширенные сетевые возможности.

2. Начало работы с "Axiom Trusted Images"

2.1. Системные и программные требования

Для работы с образами Axiom Trusted Images 25 LTS нужны:

- Процессор с 64-битной архитектурой aarch64, x86_64 и поддержкой виртуализации.
- Оперативная память: обратитесь к документации вашей ОС. Как правило, необходимый минимум - 4 ГБ RAM.
- Операционная система:
  - Linux (рекомендуемая среда; поддерживаются основные дистрибутивы: Ubuntu, CentOS, Debian и т.д.);
  - Windows (с WSL 2 или Hyper-V);
  - macOS (с Hypervisor.framework или через Docker Desktop).
- Место на диске: достаточно свободного пространства для хранения образов и контейнеров (зависит от количества и размера образов).
- Установленный Docker Engine, Docker Desktop или утилита командной строки Podman.

2.2. Создание контейнера и запуск

Поскольку образы уже загружены, то, чтобы создать контейнер из образа, выполните следующую команду.

Запустите контейнер с помощью команды docker run:

```
docker container run --rm -it <URL адрес репозитория>/<имя репозитория>:<тег образа>
```

Примечание:

Образы поставляются только в версии `latest`, что означает использование последних

версий компонентов (например Axiom JDK) и libc библиотеки musl. Если часть `:<тег образа>` опущена, она заменяется на `:latest`, то есть команда работает с тегом `latest`.

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

```
docker run -it --rm -v /home/user/project/:/data <URL адрес репозитория>/<имя репозитория>:<тег образа> java -jar /data/MyApp.jar
```

2.3. Образы и работа с ними

2.3.1. 25-trusted-base

Это базовый образ с минимальным набором компонент ОС для запуска статических слинкованных программ, либо программ, зависящих только от стандартной библиотеки C.

Примеры использования этого образа приведены в последующих разделах.

2.3.2. 25-trusted-axiom-runtime-container-pro

Данный образ содержит Axiom JDK Pro 27.

Ниже представлен пример Dockerfile, в котором Java приложение компилируется в образе 25-trusted-axiom-runtime-container-pro:

```
FROM <...>/25-trusted-axiom-runtime-container-pro:latest AS builder

COPY <<'EOF' HelloAxiom.java
public class HelloAxiom {
    public static void main(String[] args) {
        System.out.println("Hello Axiom!");
    }
}
EOF
```

2.3.3. 25-trusted-axiom-runtime-container-certified

Данный образ поставляется с Axiom JDK Certified 21.

Работа с образом 25-trusted-axiom-runtime-container-certified осуществляется аналогично

образу 25-trusted-axiom-runtime-container-pro.

2.3.4. 25-trusted-axiom-runtime-container-express

Репозиторий содержит образ с Axiom JDK Express 11.

Работа с образом 25-trusted-axiom-runtime-container-express осуществляется аналогично образу 25-trusted-axiom-runtime-container-pro.

2.3.5. 25-trusted-axiom-native-image-kit-container

Репозиторий содержит образ с Axiom NIK 25.

Ниже представлен пример Dockerfile, в котором Java приложение с помощью NIK компилируется в бинарный файл, который затем запускается в базовом образе:

```
FROM <...>/25-trusted-axiom-native-image-kit-container:latest AS builder

COPY <<'EOF' HelloAxiom.java
public class HelloAxiom {
    public static void main(String[] args) {
        System.out.println("Hello Axiom!");
    }
}
EOF

RUN javac HelloAxiom.java
RUN native-image --no-fallback -o hello-axiom HelloAxiom

FROM <...>/25-trusted-base:latest

COPY --from=builder /home/appuser/hello-axiom .

ENTRYPOINT [ "/home/appuser/hello-axiom" ]
```

2.3.6. 25-trusted-gcc

Репозиторий содержит образ с инструментами для C/C++ разработки на базе GCC 14.3.

Ниже представлен пример Dockerfile со сборкой C приложения с помощью образа 25-trusted-gcc с последующим запуском в базовом образе:

```
FROM <...>/25-trusted-gcc:latest AS builder
```

```
COPY <<'EOF' hello_axiom.c
#include <stdio.h>

int main(int argc, char *argv[]) {
    printf("Hello Axiom!\n");
}
EOF

RUN gcc -Wall -o hello-axiom hello_axiom.c

FROM <...>/25-trusted-base:latest

COPY --from=builder /home/appuser/hello-axiom .

ENTRYPOINT [ "/home/appuser/hello-axiom" ]
```

2.3.7. 25-trusted-go

Репозиторий содержит образ с установленным компилятором Go и вспомогательными утилитами и библиотеками. Образ предназначен для сборки и отладки программ написанных на Go.

Образ поставляется только с версией Go, поддерживаемой разработчиками языка Go (https://go.dev).

Ниже представлен пример Dockerfile со сборкой Go приложения с помощью образа 25-trusted-go с последующим запуском в базовом образе:

```
FROM <...>/25-trusted-go:latest AS builder

COPY <<'EOF' hello_axiom.go
package main

import "fmt"

func main() {
    fmt.Println("Hello Axiom")
}
EOF

RUN go build -o hello-axiom hello_axiom.go

FROM <...>/25-trusted-base:latest
```

```
COPY --from=builder /home/appuser/hello-axiom .

ENTRYPOINT [ "/home/appuser/hello-axiom" ]
```

2.3.8. 25-trusted-python

Репозиторий содержит образ с Python 3.12 и базовыми Python утилитами (pip, setuptools, wheel).

Пример Dockerfile с запуском приложения в образе 25-trusted-python:

```
FROM <...>/25-trusted-python:latest

COPY <<'EOF' hello_axiom.py
if __name__ == "__main__":
    print("Hello Axiom")
EOF

CMD [ "hello_axiom.py" ]
```

2.3.9. 25-trusted-nginx

Репозиторий содержит образ с веб-сервером Nginx и содержит минимальный набор пакетов, необходимых для работы `nginx`. Не содержит никаких динамических модулей `nginx`.

Следующий пример кода демонстрирует Dockerfile, который создает образ для размещения простого статического контента:

```
FROM 25-trusted-nginx:latest
COPY static-html-directory /var/lib/nginx/html
```

Выполните следующие команды в директории, где находится директория `static-html-directory`, чтобы собрать и запустить контейнер:

```
$ docker build -t my-nginx .
$ docker run -d -p 8080:8080 my-nginx
```

После этого вы сможете получить доступ к http://localhost:8080.

2.3.10. 25-trusted-node

Репозиторий содержит образы с Node.js и базовыми утилитами (npm):

Образ поставляется только с версией Node.js, поддерживаемой разработчиками Node.js (https://nodejs.org).

Для корректной обработки сигналов ОС рекомендуется в Dockerfile запускать процесс `node` не напрямую, а используя прослойку `/usr/sbin/tini`, которая уже предустановлена во всех образах.

Пример Dockerfile с запуском приложения в образе 25-trusted-node:

```
FROM <...>/25-trusted-node:latest

COPY <<'EOF' hello_axiom.js
import { createServer } from 'node:http';

const server = createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello Axiom!\n');
});

server.listen(5000, '0.0.0.0', () => {
  console.log('Listening on 0.0.0.0:5000');
});
EOF

ENTRYPOINT [ "/usr/sbin/tini", "--"]
CMD [ "node", "hello_axiom.js"]
```

AXIOM JDK

Axiom Trusted Images 25 LTS
Руководство пользователя