



Java-разработчик

Раньше программировал на XML,
теперь больше на YAML.

О чем сегодня будем говорить?

GraalVM

native image

Где это ваш graalvm может быть еще полезен?

GraalVM

native image



О чем сегодня будем говорить?

GraaIVM

native image

В предыдущих сериях..



GraalVM Native Image: польза в реальной жизни



Андрей
Чухлебов

Сбер

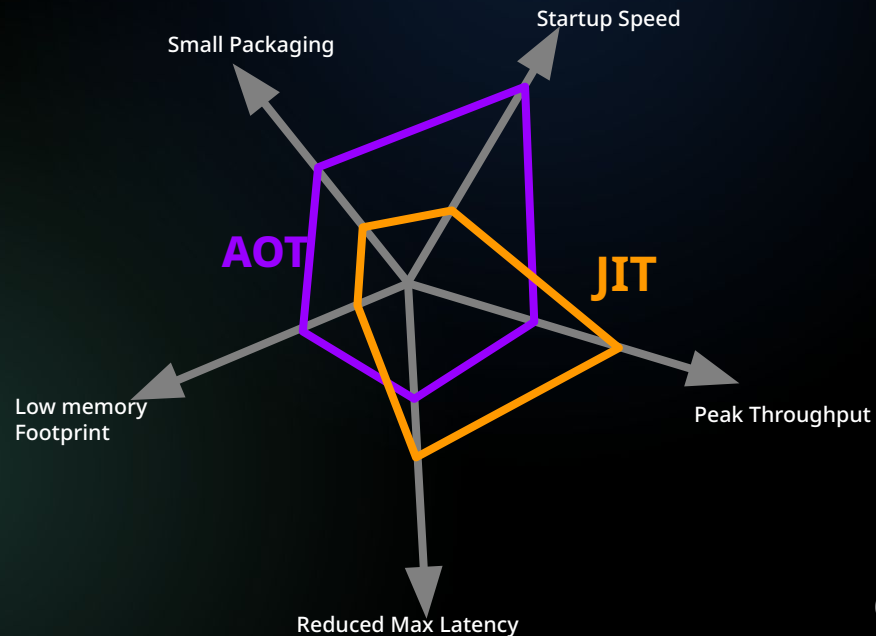


<https://www.youtube.com/watch?v=H3iLihCxpw8&t=158s>

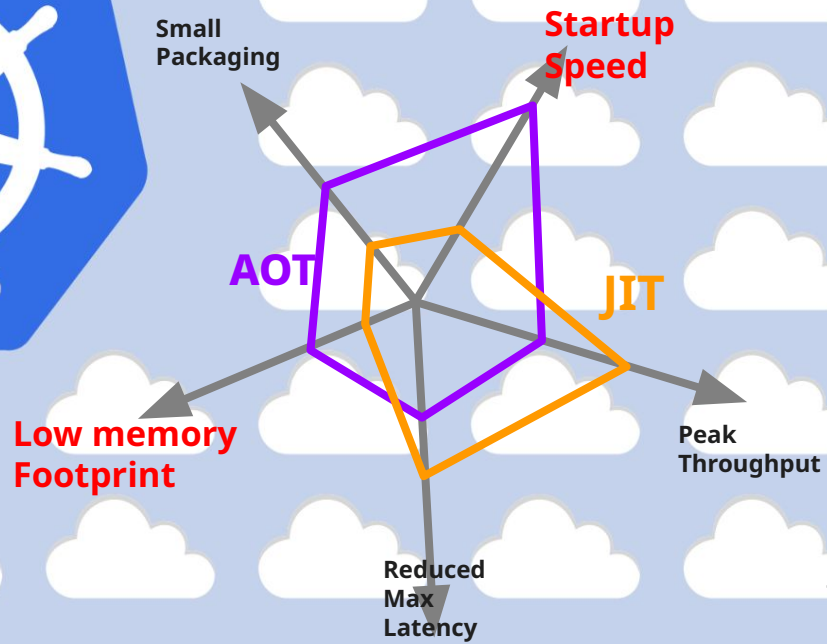
Где это ваш graalvm может быть еще полезен?

GraalVM

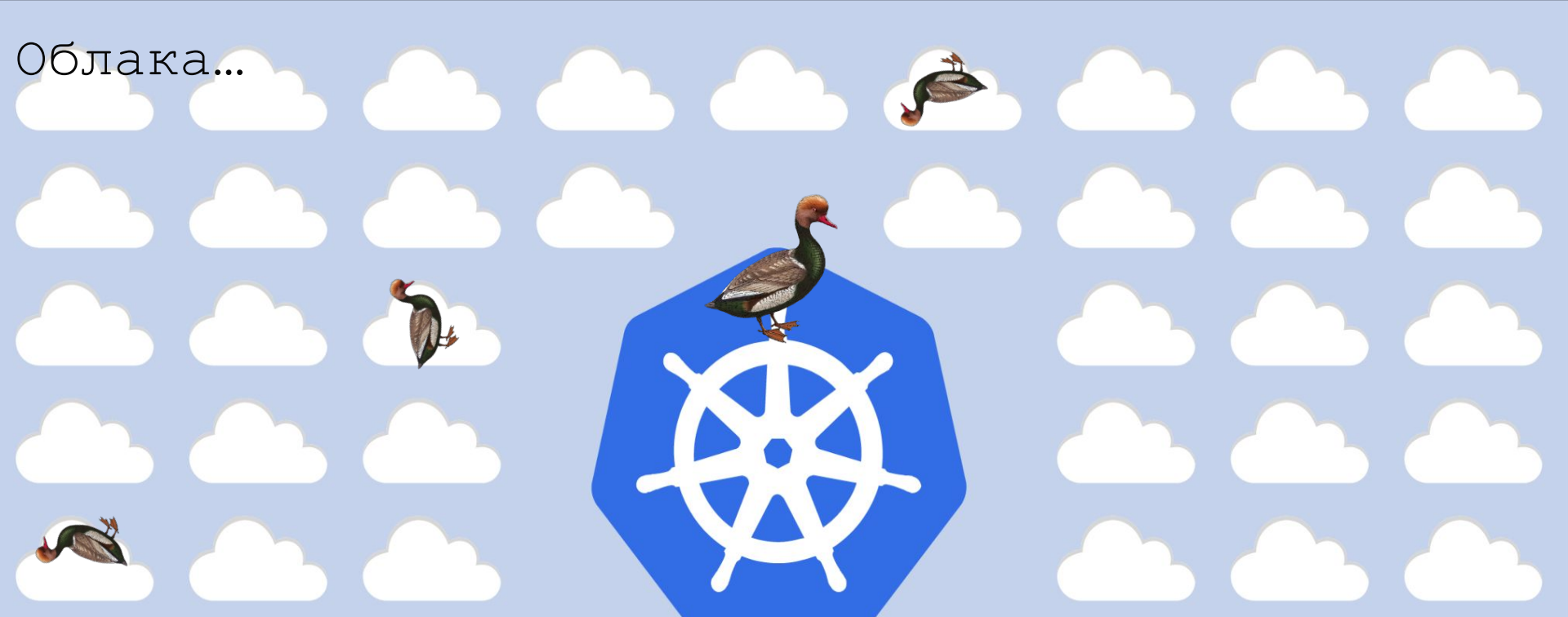
native image



Облака...



Облака...

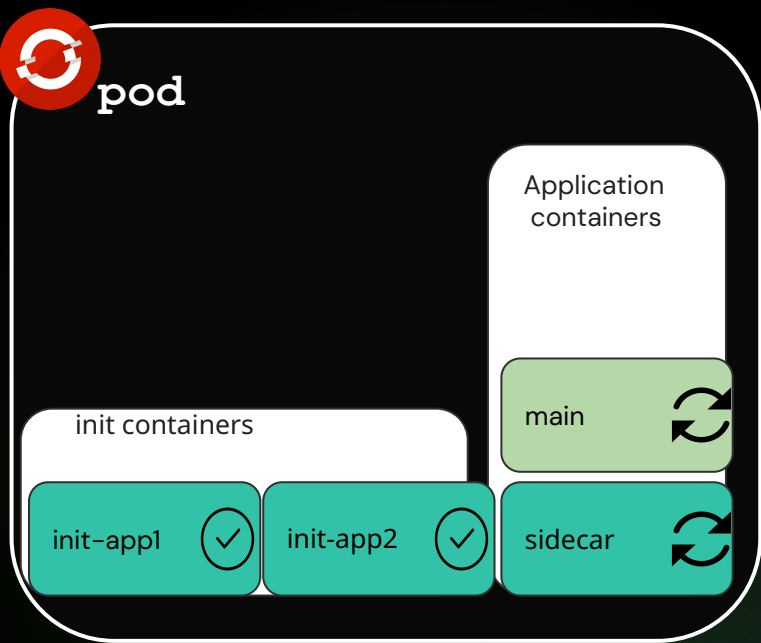


O'REILLY®

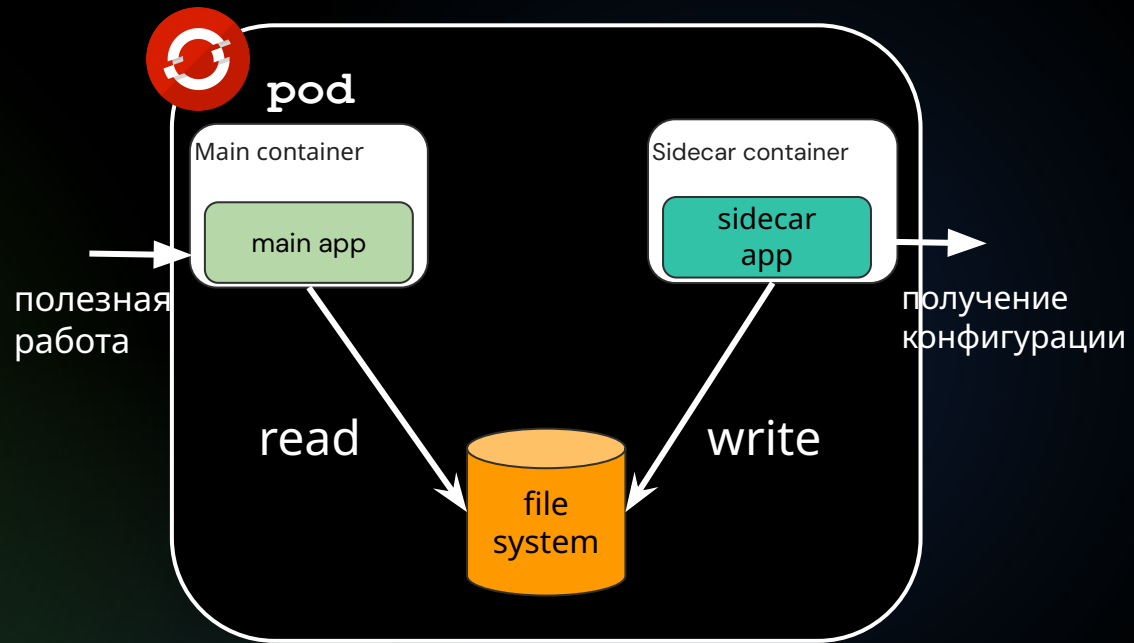
Kubernetes Patterns

Reusable Elements for Designing
Cloud-Native Applications

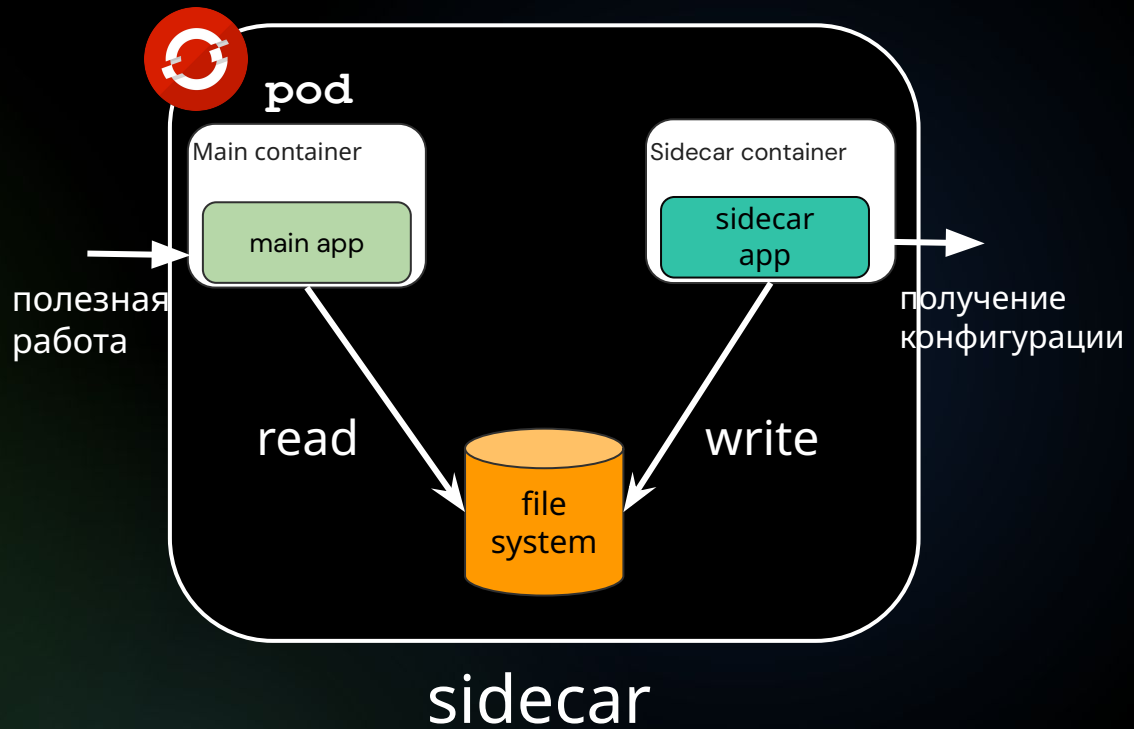
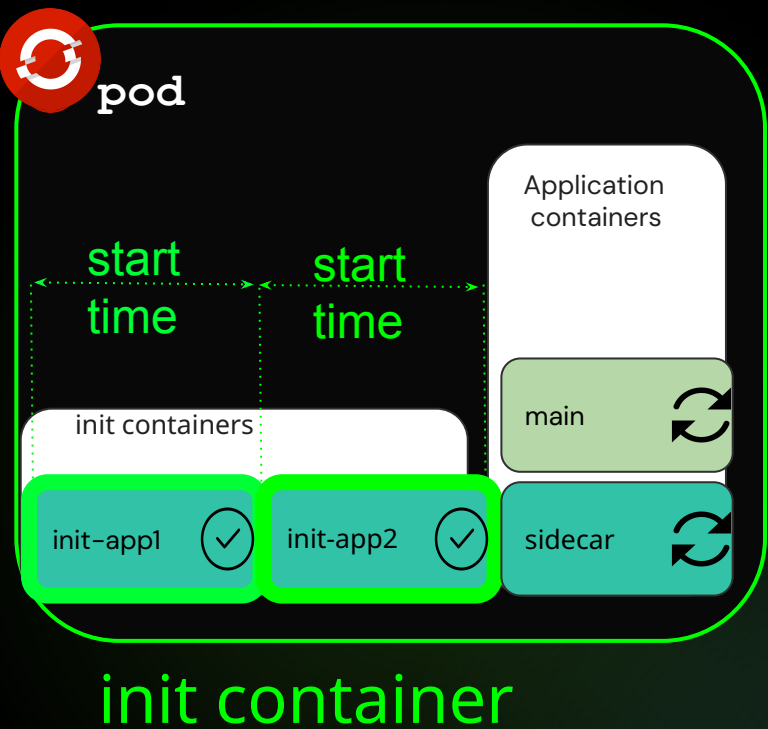


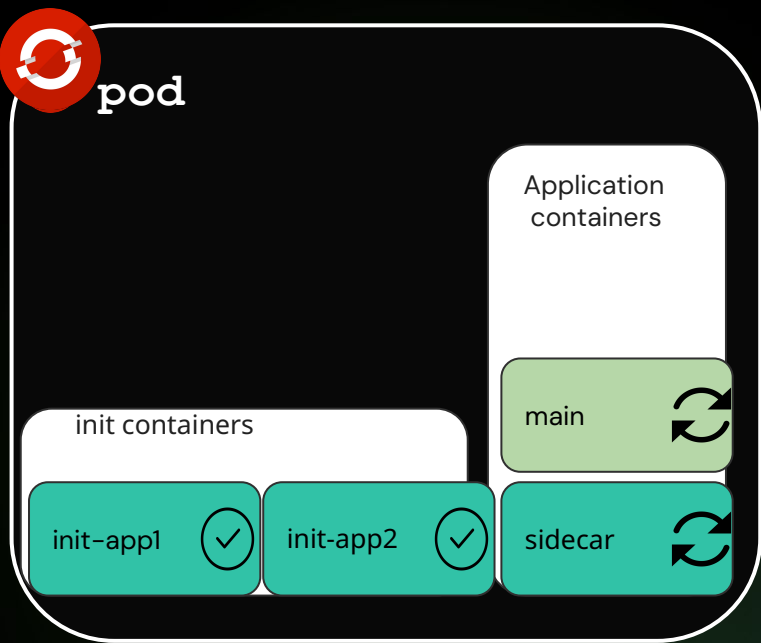


init container

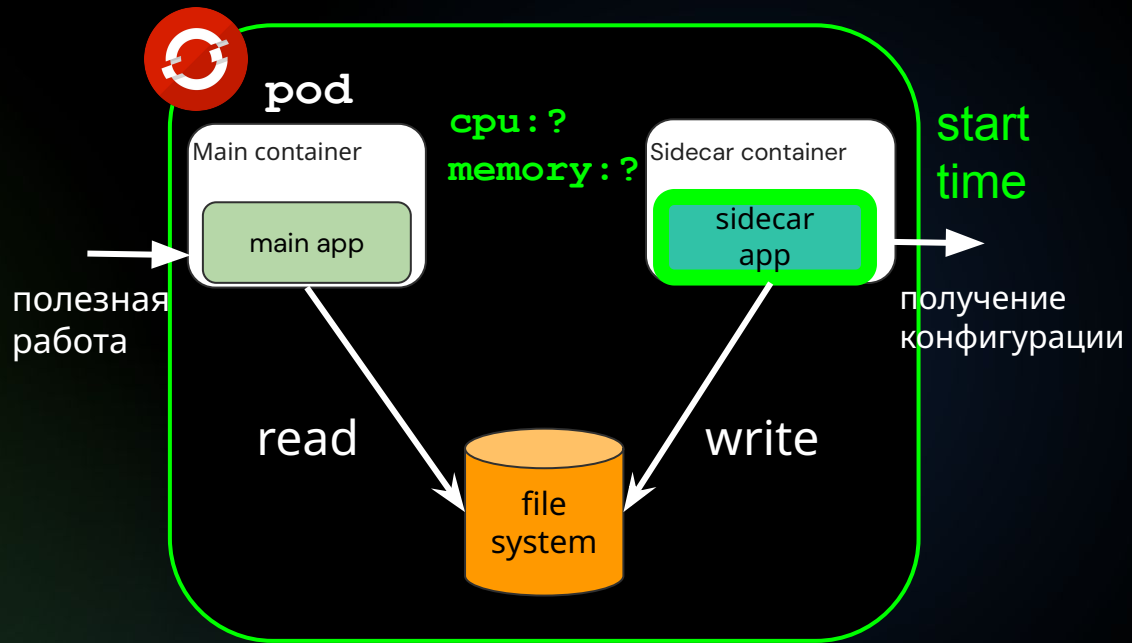


sidecar





init container



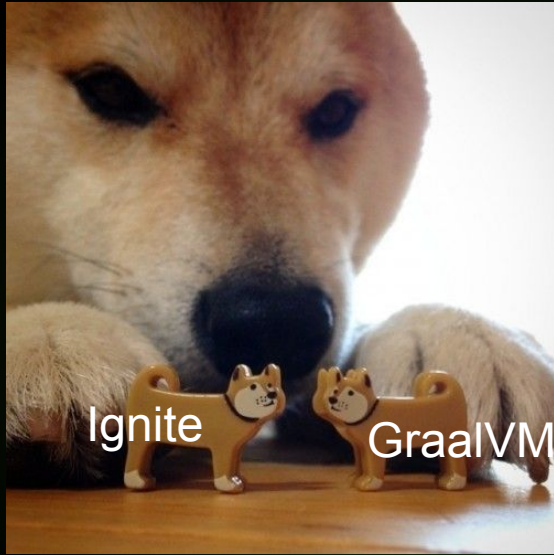
sidecar

В предыдущих сериях..

- CRaC
- JitServer от openj9
- Игры с флажками и тд
- Class Data Sharing
- ...



Ignite + GraalVM



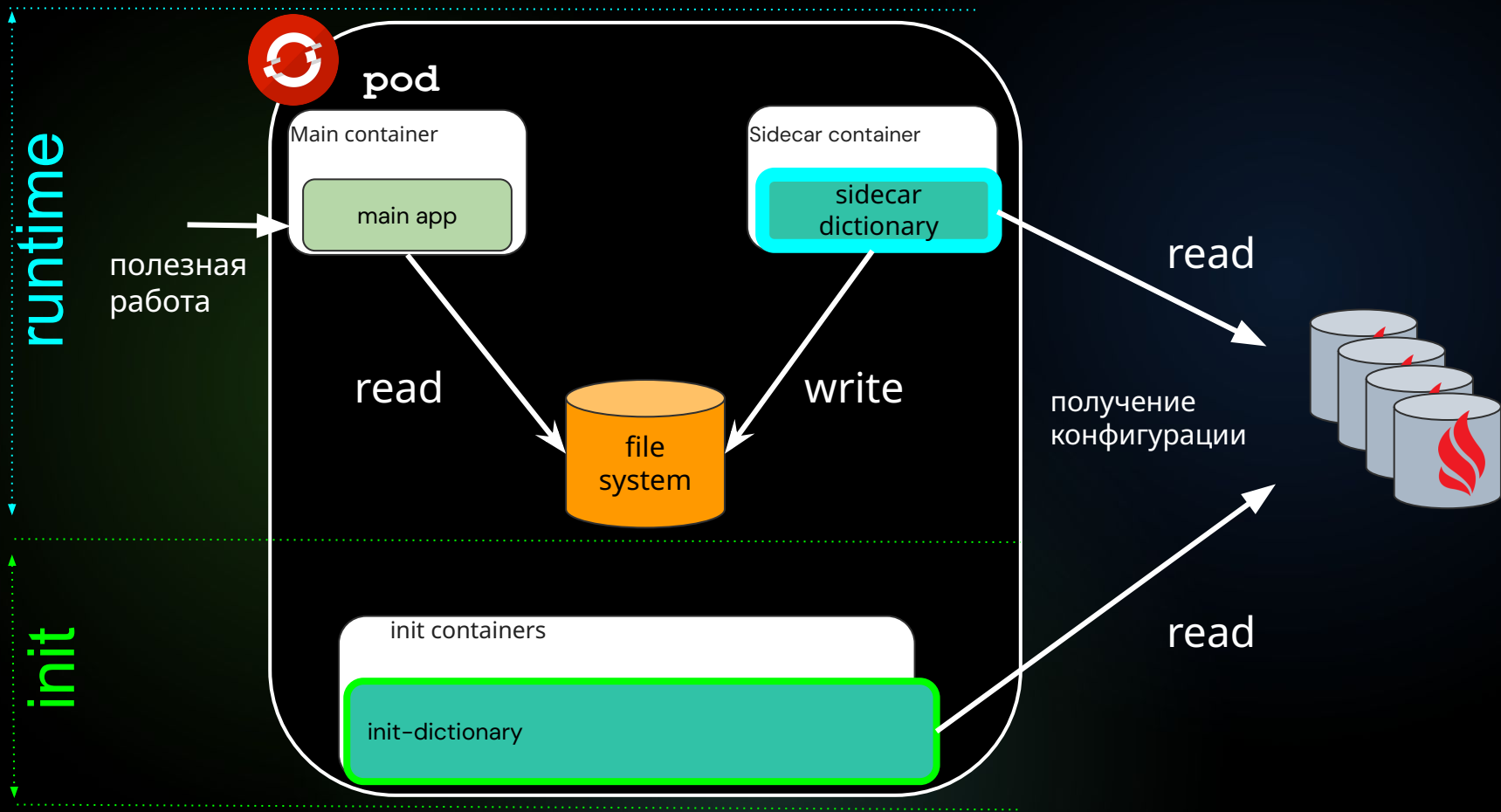
apache

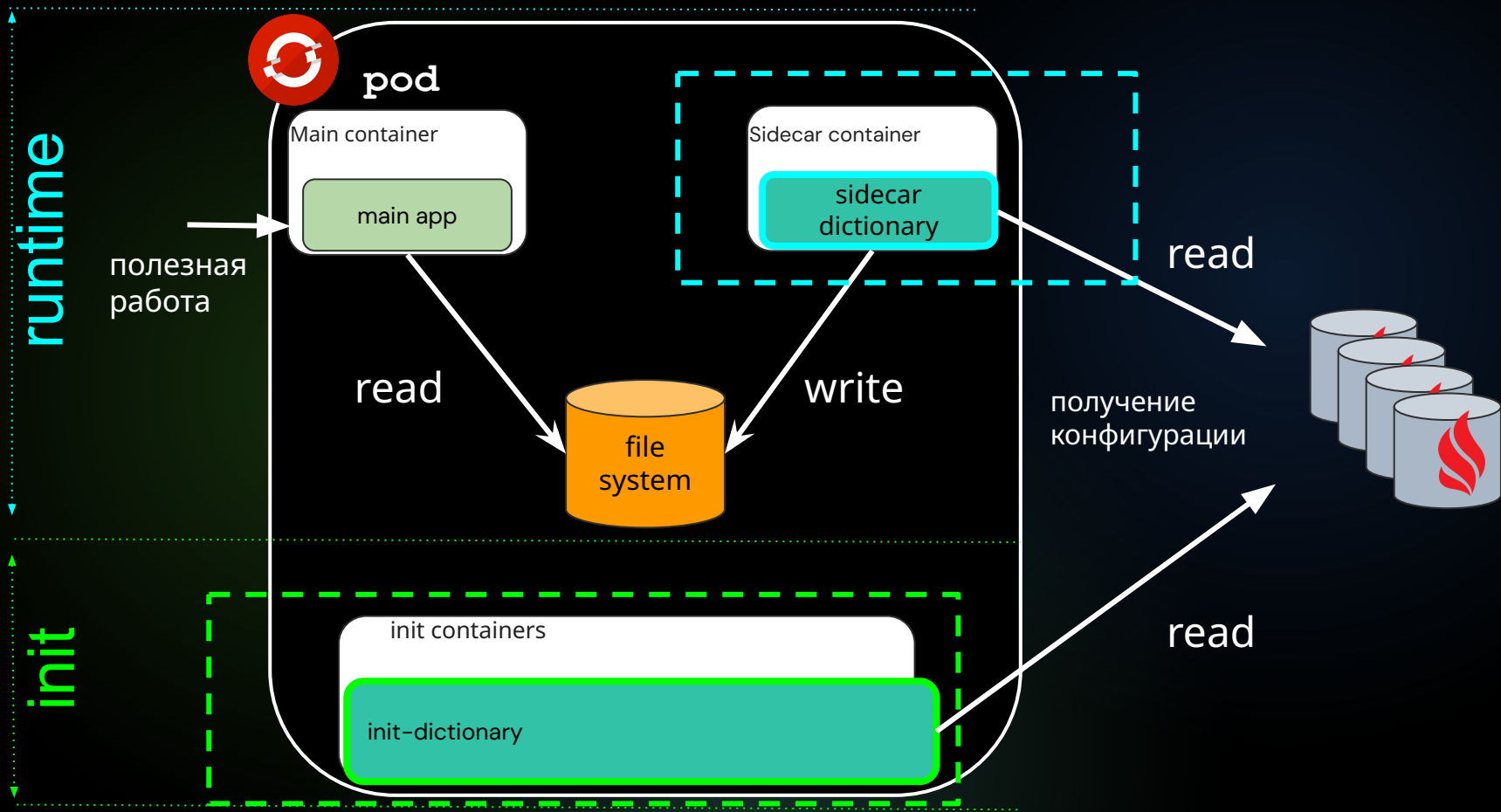
Ignite

+

GraalVM

native image





dependency

```
<dependency>  
  <groupId>org.apache.ignite</groupId>  
  <artifactId>ignite-core</artifactId>  
  <version>${ignite.version}</version>  
</dependency>
```


mvn dependency:tree

```
[INFO] --- dependency:3.7.0:tree (default-cli) @ joker-ignite ---  
[INFO] org.example:joker-ignite:jar:1.1-SNAPSHOT  
[INFO] +- org.apache.ignite:ignite-core:jar:2.16.0:compile  
[INFO] | +- javax.cache:cache-api:jar:1.0.0:compile  
[INFO] | \- org.jetbrains:annotations:jar:16.0.3:compile
```

```
String mode = System.getProperty("mode", "init");
ClientConfiguration cfg = new ClientConfiguration().setAddresses("host:port");
IgniteClient client = Ignition.startClient(cfg);
ClientCache<String, String> cache = client.cache("JOKER_CACHE");
if (mode.equals("init")) {
    Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    System.exit(0);
} else if (mode.equals("sidecar")) {
    while (true) {
        Thread.sleep(1000);
        Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    }
}
```

```
String mode = System.getProperty("mode", "init");
ClientConfiguration cfg = new ClientConfiguration().setAddresses("host:port");
IgniteClient client = Ignition.startClient(cfg);
ClientCache<String, String> cache = client.cache("JOKER_CACHE");
if (mode.equals("init")) {
    Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    System.exit(0);
} else if (mode.equals("sidecar")) {
    while (true) {
        Thread.sleep(1000);
        Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    }
}
```

```
String mode = System.getProperty("mode", "init");
ClientConfiguration cfg = new ClientConfiguration().setAddresses("host:port");
IgniteClient client = Ignition.startClient(cfg);
ClientCache<String, String> cache = client.cache("JOKER_CACHE");
if (mode.equals("init")) {
    Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    System.exit(0);
} else if (mode.equals("sidecar")) {
    while (true) {
        Thread.sleep(1000);
        Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    }
}
```

```
String mode = System.getProperty("mode", "init");
ClientConfiguration cfg = new ClientConfiguration().setAddresses("host:port");
IgniteClient client = Ignition.startClient(cfg);
ClientCache<String, String> cache = client.cache("JOKER_CACHE");
if (mode.equals("init")) {
    Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    System.exit(0);
} else if (mode.equals("sidecar")) {
    while (true) {
        Thread.sleep(1000);
        Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    }
}
```

```
String mode = System.getProperty("mode", "init");
ClientConfiguration cfg = new ClientConfiguration().setAddresses("host:port");
IgniteClient client = Ignition.startClient(cfg);
ClientCache<String, String> cache = client.cache("JOKER_CACHE");
if (mode.equals("init")) {
    Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    System.exit(0);
} else if (mode.equals("sidecar")) {
    while (true) {
        Thread.sleep(1000);
        Utils.writeFile(cache.get("JOKER_DICTIONARY"));
    }
}
```

Пакуем в контейнер

```
FROM openjdk:17-jdk-alpine AS build
WORKDIR /app
COPY pom.xml .
COPY src ./src
COPY mvnw .
COPY .mvn .mvn
RUN ./mvnw clean package
```

```
FROM openjdk:17-jdk-alpine
WORKDIR /app
COPY --from=build joker-ignite-1.0.jar ./app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

```
FROM openjdk:17-jdk-alpine AS build
WORKDIR /app
COPY pom.xml .
COPY src ./src
COPY mvnw .
COPY .mvn .mvn
RUN ./mvnw clean package
```

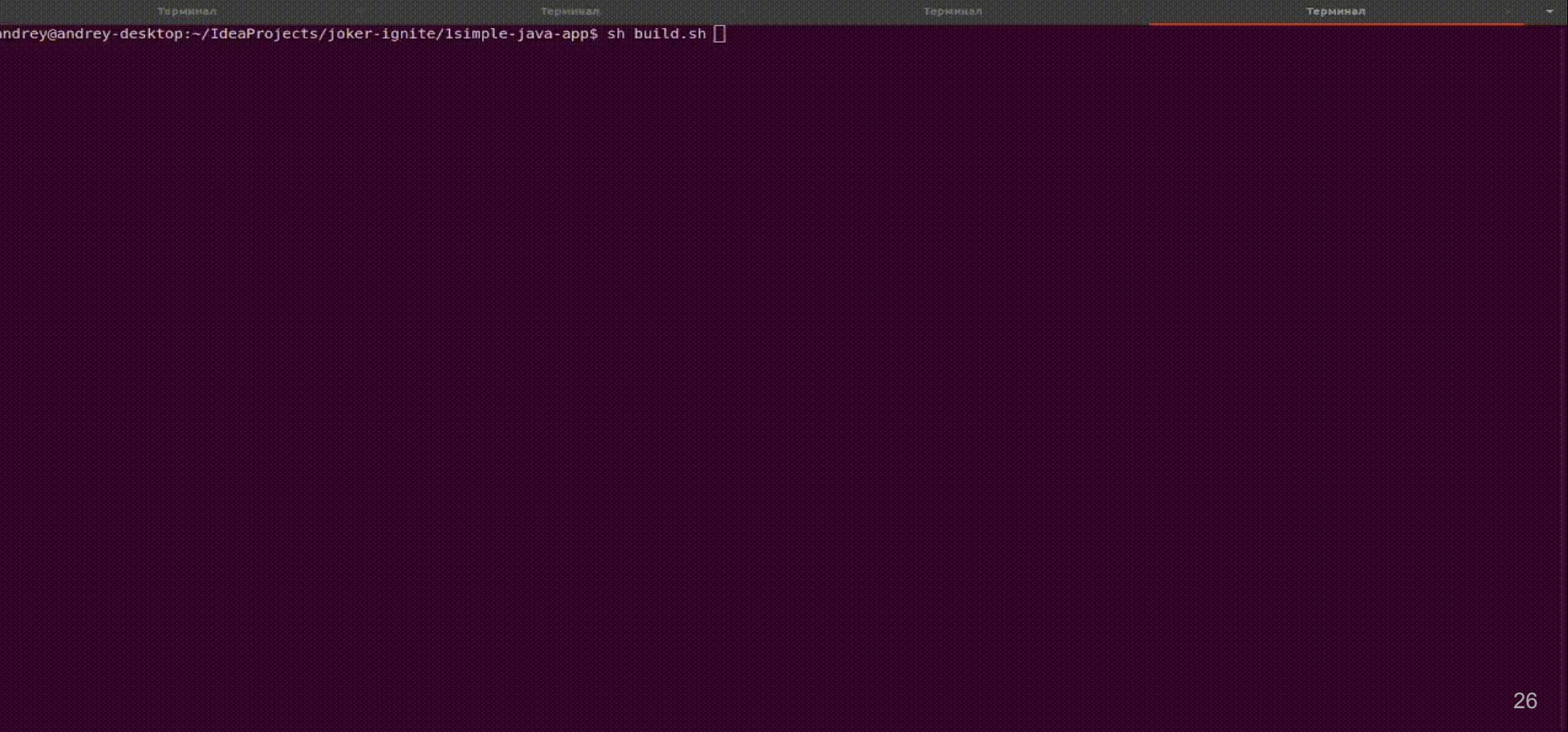
```
FROM openjdk:17-jdk-alpine
WORKDIR /app
COPY --from=build joker-ignite-1.0.jar ./app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]
```



```
FROM openjdk:17-jdk-alpine AS build
WORKDIR /app
COPY pom.xml .
COPY src ./src
COPY mvnw .
COPY .mvn .mvn
RUN ./mvnw clean package
```

```
FROM openjdk:17-jdk-alpine
WORKDIR /app
COPY --from=build joker-ignite-1.0.jar ./app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

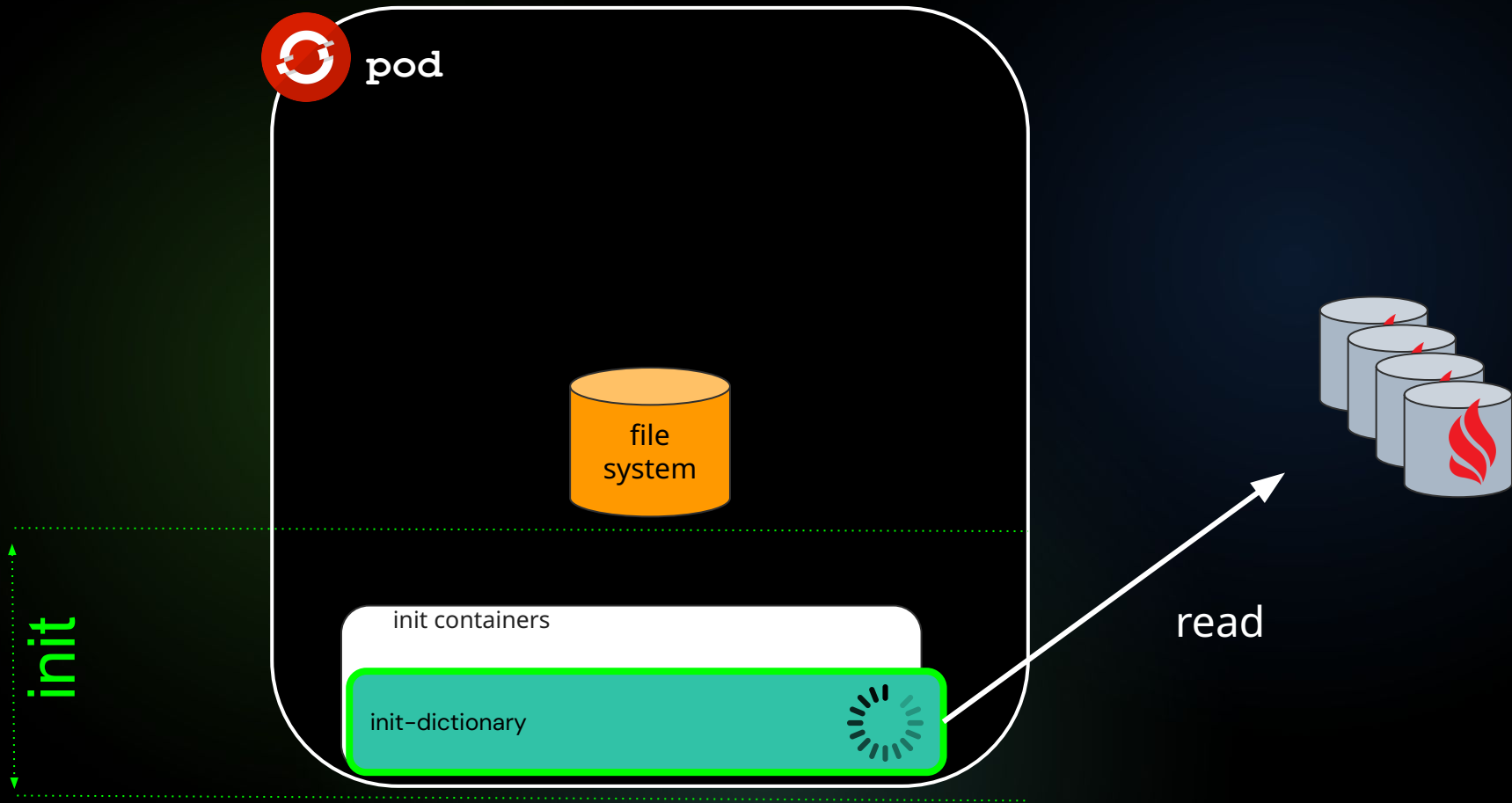
```
docker build --progress=plain -t ignite-jvm -f Dockerfile .  
docker run --rm ignite-jvm
```

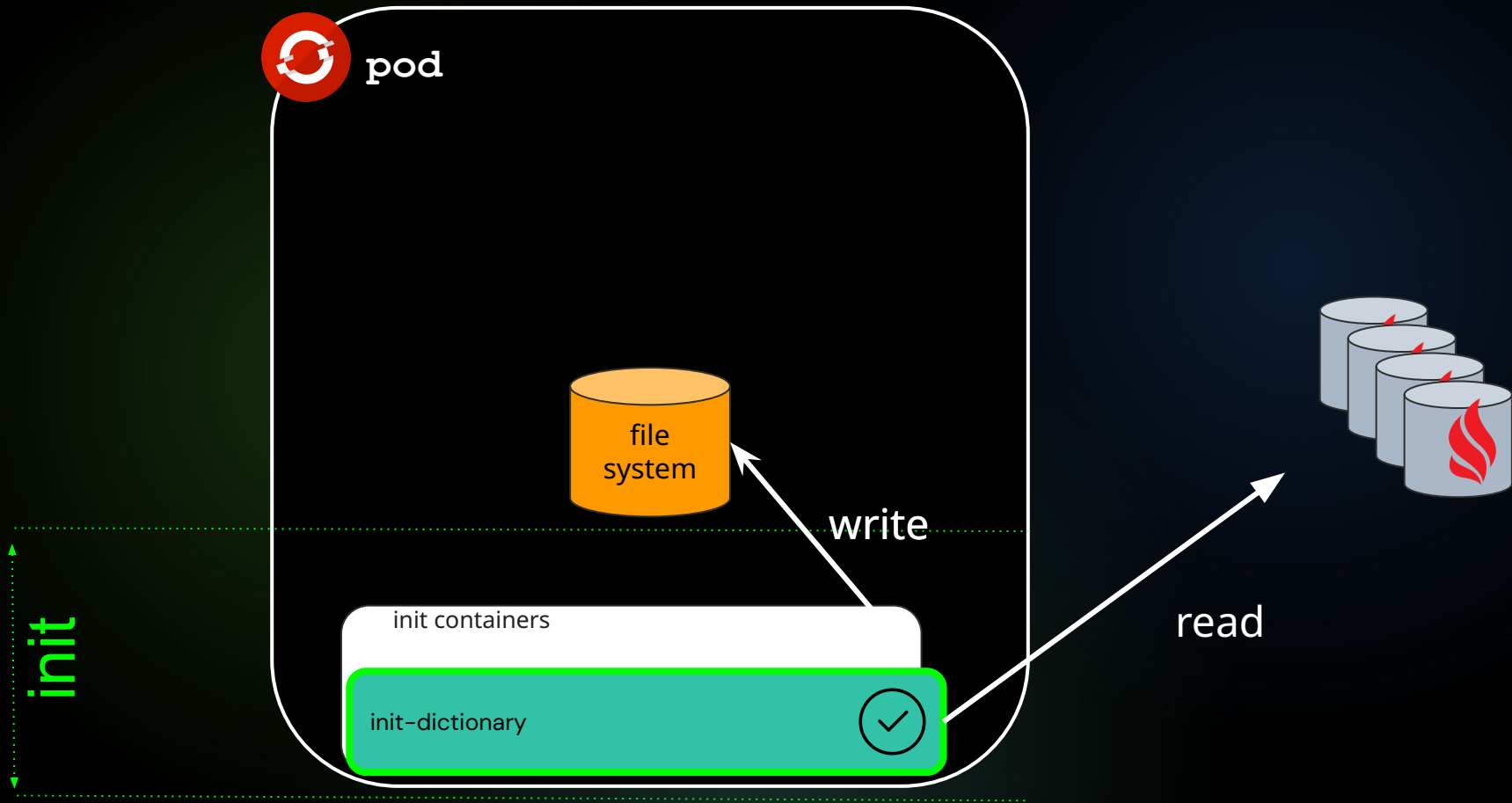


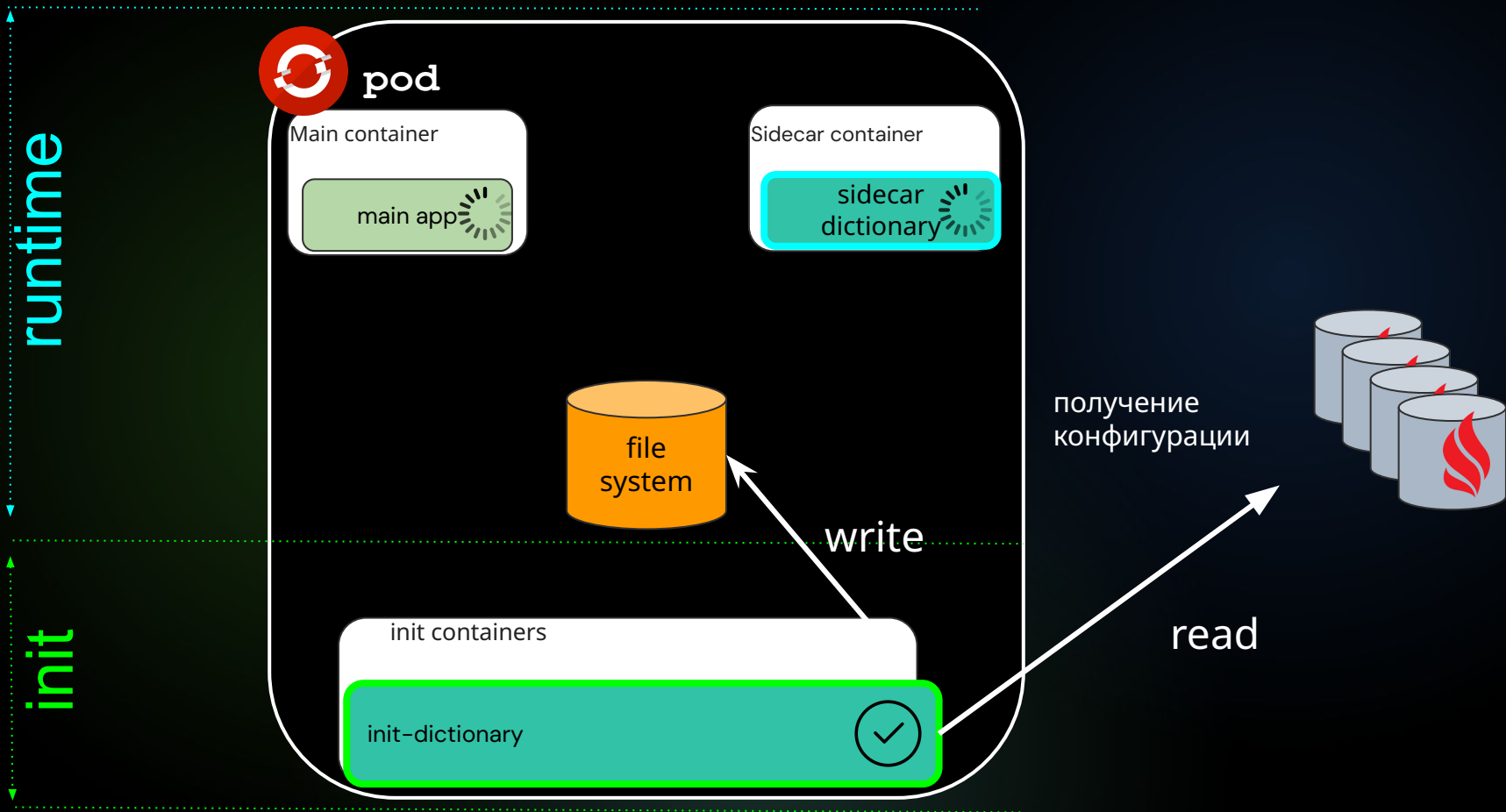


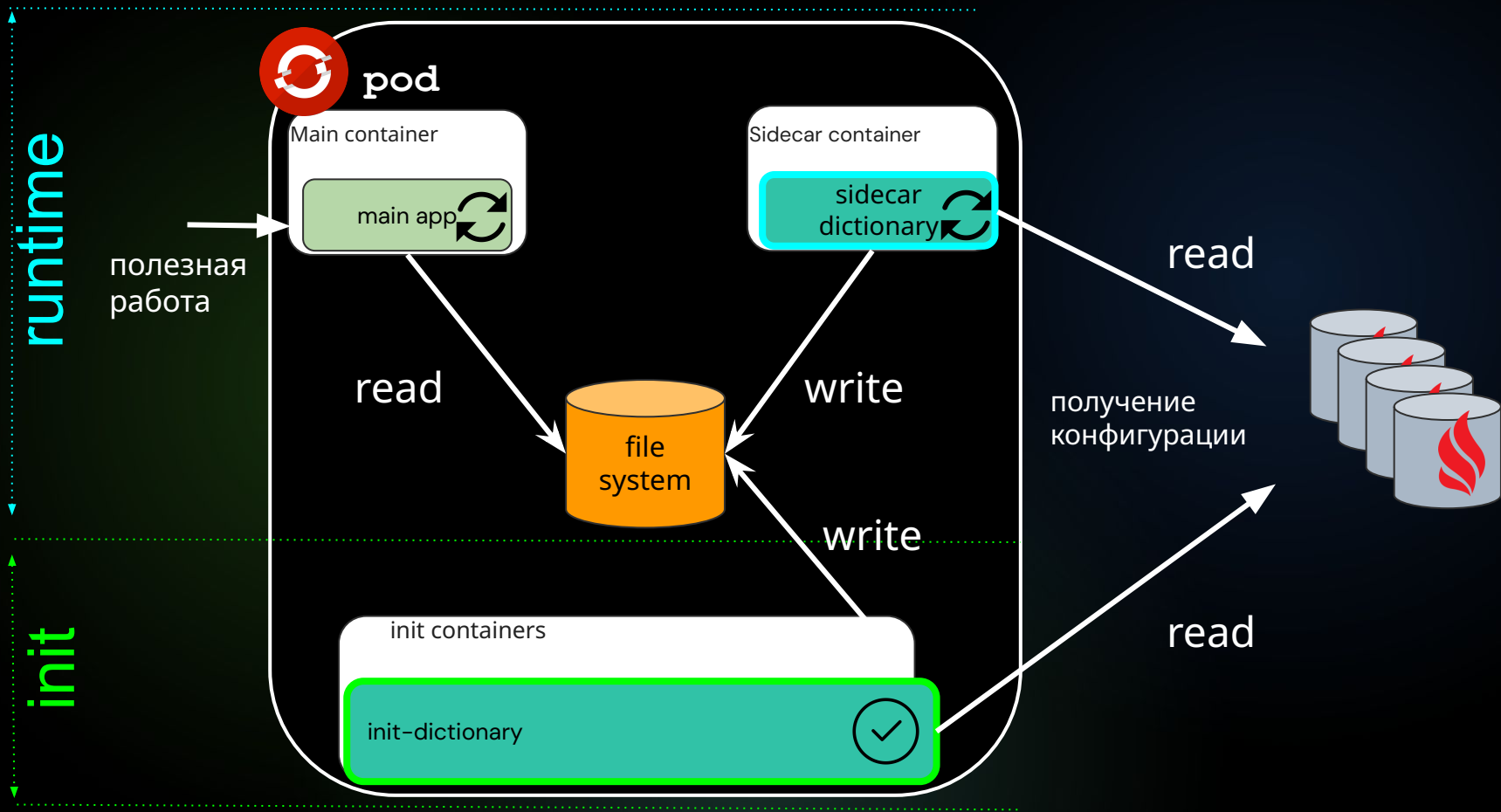
pod





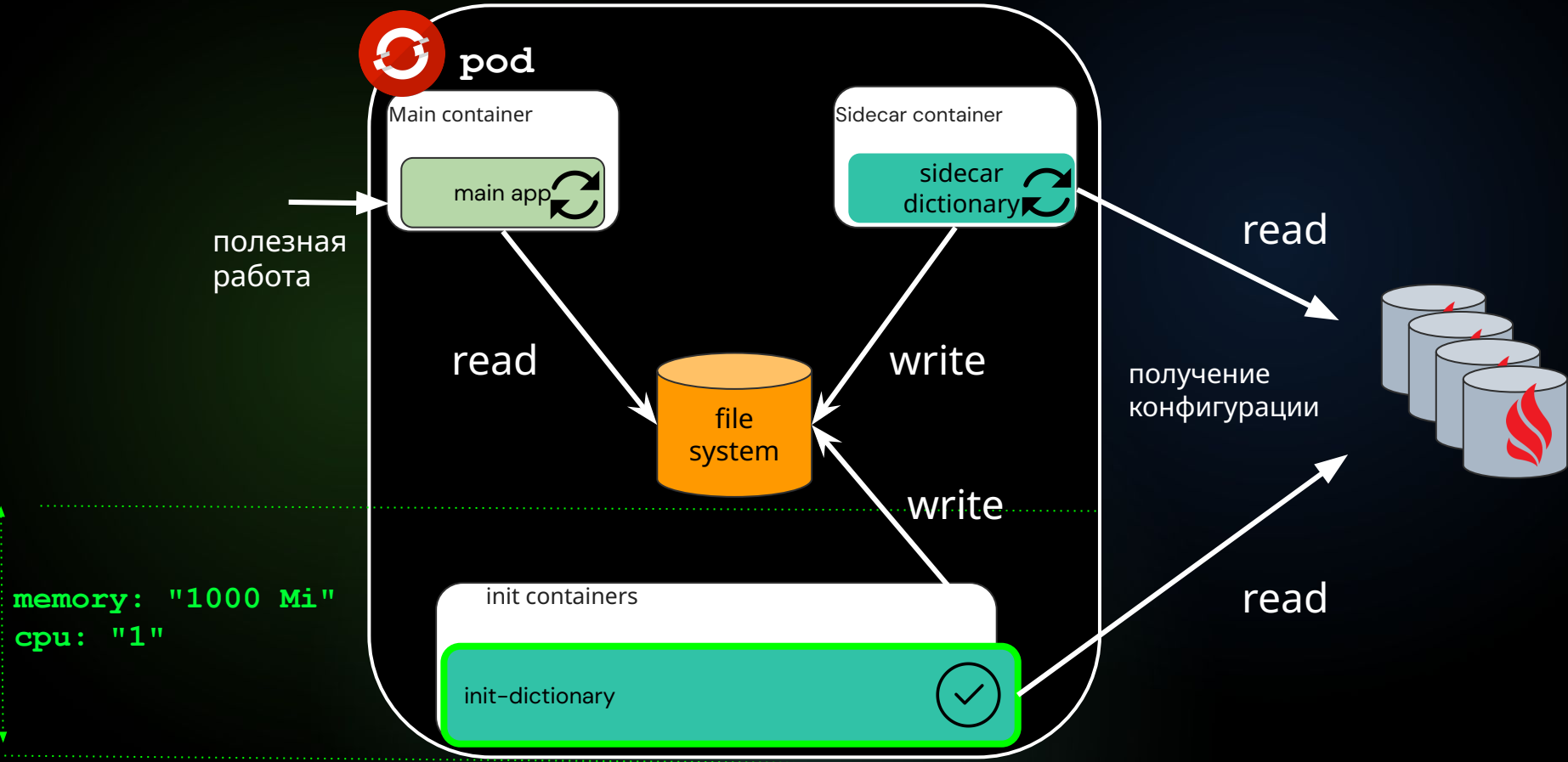


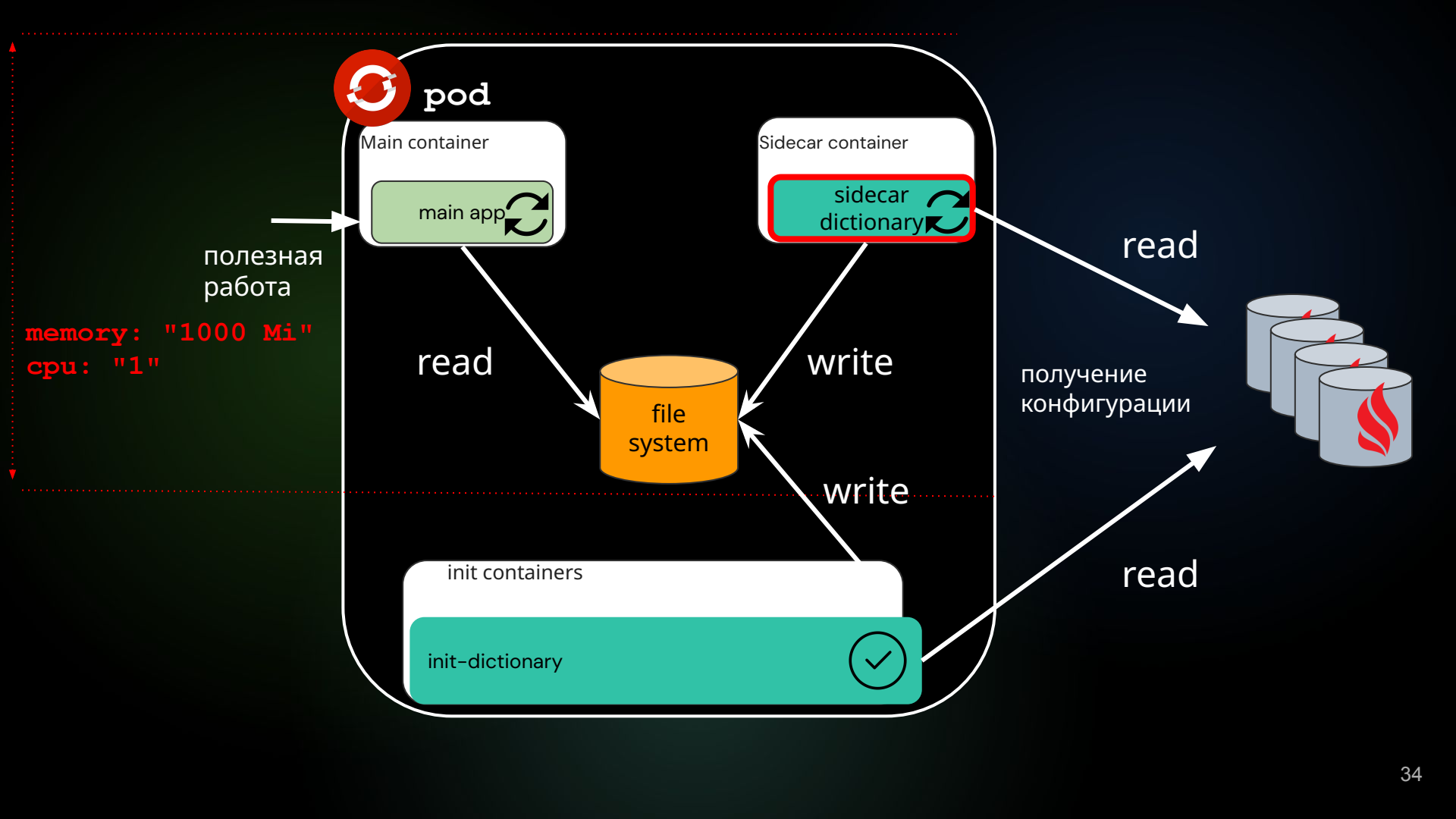




Какие проблемы?

cpu	Время старта
1	~4 s
0.5	~5 s
0.3	~15 s
0.1	~48 s
0.01	~не дождался





Попробуем собрать с помощью native-image

GraalVM

native image



<https://github.com/AndreyChukhlebov/joker-ignite>

Попробуем собрать с помощью native-image

```
FROM openjdk:17-jdk-alpine AS build
WORKDIR /app
COPY pom.xml .
COPY src ./src
COPY mvnw .
COPY .mvn .mvn
RUN ./mvnw clean package
```

...



Собираем jar

Попробуем собрать с помощью native-image

```
FROM openjdk:17-jdk-alpine AS build
```

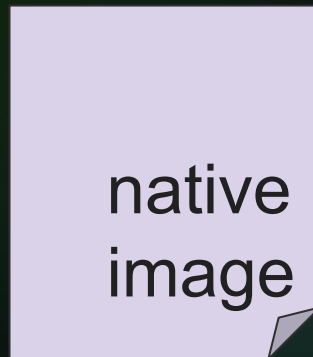
```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
```

```
WORKDIR /app
```

```
COPY --from=build target/joker-ignite-1.0.jar ./app.jar
```

```
RUN native-image -jar app.jar
```

...



Собираем jar

Собираем native image

Попробуем собрать с помощью native-image

```
FROM openjdk:17-jdk-alpine AS build
```

```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
```

```
FROM alpine:3.14
```

```
WORKDIR /app
```

```
COPY --from=build-native /app /app
```

```
ENTRYPOINT ["/app"]
```

Попробуем собрать с помощью native-image

```
FROM openjdk:17-jdk-alpine AS build
```

```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
```

```
FROM alpine:3.14
```

```
docker build native:1 -f Dockerfile .
```

```
docker run native:1
```

Сразу так много вопросов и так мало ответов.

```
Терминал Терминал Терминал Терминал
andrey@andrey-desktop:~/IdeaProjects/joker-ignite/4.0native-fallnack$ sh build.sh
Error response from daemon: No such image: native_0_1:1
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 1.97kB done
#1 DONE 0.0s

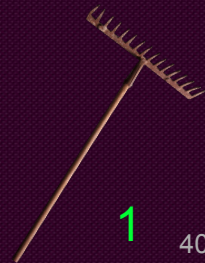
#2 [internal] load metadata for docker.io/library/m2image:latest
#2 DONE 0.0s

#3 [internal] load metadata for ghcr.io/graalvm/native-image-community:17
#3 DONE 0.0s

#4 [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine
#4 DONE 0.7s

#5 [internal] load metadata for docker.io/library/alpine:3.14
□
```

 **exec ./app: not found**




```
FROM ghcr.io/graalvm/native-image-community:17
```

```
RUN ldd --version
```

```
ldd (GNU libc) 2.34
```

```
FROM alpine:3.14
```

```
RUN ldd --version
```

```
musl libc (x86_64)
```

```
Version 1.2.2
```

```
FROM ghcr.io/graalvm/native-image-community:17
```

```
RUN ldd --version
```

```
ldd (GNU libc) 2.34
```

```
FROM alpine:3.14
```

```
RUN ldd --version
```

```
musl libc (x86_64)
```

```
Version 1.2.2
```

```
FROM ghcr.io/graalvm/native-image-community:17  
RUN ldd --version  
  
ldd (GNU libc) 2.34
```

```
FROM alpine:3.14  
RUN ldd --version  
  
musl libc (x86_64)  
Version 1.2.2
```

Решения

1. Добавить glibc

```
FROM alpine:3.14
```

```
RUN apk update && apk add --no-cache gcompat libstdc++
```

Решения

2. Запустить на сборочном образе или там, где есть все библиотеки

```
FROM ghcr.io/graalvm/native-image-community:17
```

```
COPY --from=build-native /app /app
```

```
ENTRYPOINT ["/app"]
```



Ваш иماج

~800 MB

Решения

3. —static

...

```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
WORKDIR /app
COPY --from=build target/joker-ignite-1.1.jar ./app.jar
RUN native-image -static -jar app.jar
```

Пробуем дальше...

```
...  
FROM ghcr.io/graalvm/native-image-community:17 as build-native  
WORKDIR /app  
COPY --from=build_target/joker-ignite-1.1-SNAPSHOT.jar ./app.jar  
RUN native-image -static -jar app.jar  
...
```

```
FROM alpine:3.14          FROM scratch          FROM утюг  
ENTRYPOINT ["./app"]    ENTRYPOINT ["./app"]    ENTRYPOINT ["./app"]  
  
Запускается             Запускается             Запускается
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
native	mostly_static	b9cfcddfc10f	10 seconds ago	60.1MB
native	static	e240f3d46e7e	4 minutes ago	61.7MB

Пробуем дальше...

```
andrey@andrey-desktop:~/IdeaProjects/joker-ignite/4.1native-fallnack$ sh build.sh
Untagged: native_0 2:1
Deleted: sha256:d9f03addce9663bfc0a482ca58c3bb5eaf084d4d69627117affedf713005ef23
#0 building with "default" instance using docker driver

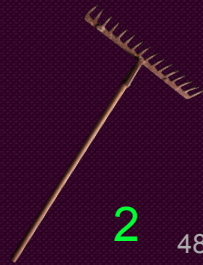
#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 2.04kB done
#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/m2image:latest
#2 DONE 0.0s

#3 [internal] load metadata for ghcr.io/graalvm/native-image-community:17
#3 DONE 0.0s

#4 [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine
□
```

 **exec ./app: No such file or directory**



Пробуем дальше...

```
Терминал Терминал Терминал Терминал
andrey@andrey-desktop:~/IdeaProjects/joker-ignite/4.1native-fallnack$ sh build.sh
Untagged: native_0 2:1
Deleted: sha256:d9f03addce9663bfc0a482ca58c3bb5eaf084d4d69627117affedf713005ef23
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 2.04kB done
#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/m2image:latest
#2 DONE 0.0s

#3 [internal] load metadata for ghcr.io/graalvm/native-image-community:17
#3 DONE 0.0s

#4 [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine
█
```

Попробуем собрать

```
FROM alpine:3.14->ghcr.io/graalvm/native-image-community:17
WORKDIR /app
COPY --from=buildnative /app /app
ENTRYPOINT ["./app"]
```

```
docker build native:1 -f Dockerfile
docker run native:1
```

Ураа! Приложение уже бросает ошибки.

```
Терминал Терминал Терминал Терминал Терминал
andrey@andrey-desktop:~/IdeaProjects/joker-ignite/4.2native-fallnack$ sh build.sh
Error response from daemon: No such image: native 0 3:1
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 2.05kB done
#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/m2image:latest
#2 DONE 0.0s

#3 [internal] load metadata for ghcr.io/graalvm/native-image-community:17
#3 DONE 0.0s

#4 [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine
#4 DONE 0.5s

#5 [internal] load .dockerignore
#5 transferring context: 2B done
#5 DONE 0.0s

#6 [buildnative 1/4] FROM ghcr.io/graalvm/native-image-community:17
#6 DONE 0.0s

#7 [m2 1/1] FROM docker.io/library/m2image:latest
#7 CACHED

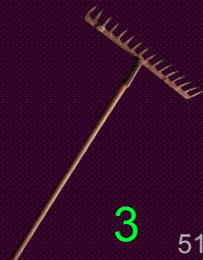
#8 [buildnative 2/4] WORKDIR /app
#8 CACHED

#9 [build 1/10] FROM docker.io/library/openjdk:17-jdk-alpine@sha256:4b6abae565492dbe9e7a894137c966a7485154238902f2f25e9dbd9784383d81
#9 CACHED

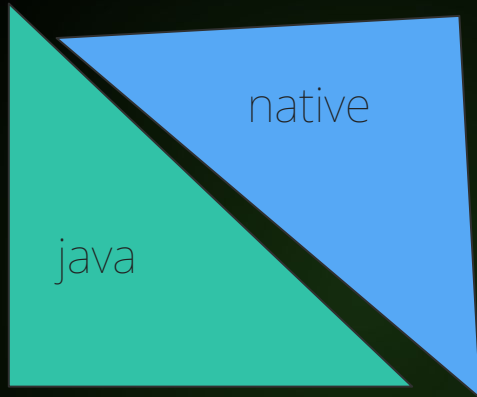
#10 [internal] load build context
#10 transferring context: 62.26kB done
#10 DONE 0.0s

#11 [stage-3 3/6] COPY target/client.jks .
#11 DONE 0.1s

#12 [build 2/10] COPY --from=m2 /root/.m2/repository /root/.m2/repository
```



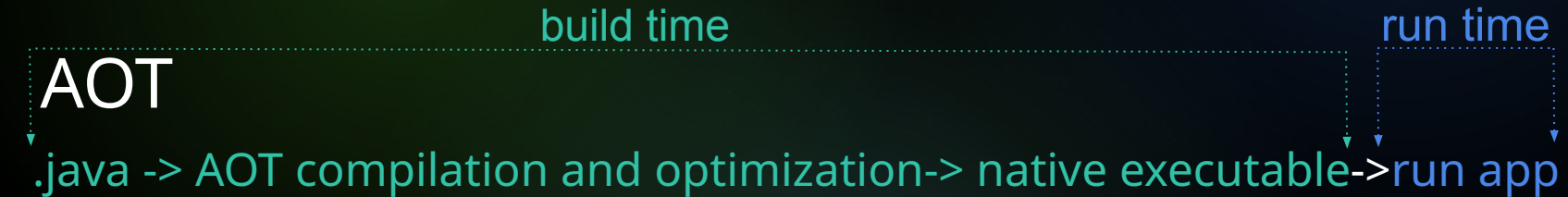
Успех!



наполовину
native



JIT VS AOT



Пробуем дальше

```
FROM ghcr.io/graalvm/native-image-community:17-> alpine:3.14
WORKDIR /app
COPY --from=buildnative /app /app
ENTRYPOINT ["/app"]
```

Пробуем дальше --no-fallback

```
Warning: Image 'app' is a fallback image that requires a JDK for  
execution
```

```
(use --no-fallback to suppress fallback image generation and to print  
more detailed information why a fallback image was necessary).
```

--initialize-at-build-time

...

```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
```

```
WORKDIR /app
```

```
COPY --from=build target/joker-ignite-1.1.jar ./app.jar
```

```
RUN native-image --static --no-fallback
```

```
--initialize-at-build-time -jar app.jar
```

...

AOT

build time

run
time

JIT

build
time

run time

`--initialize-at-build-time`

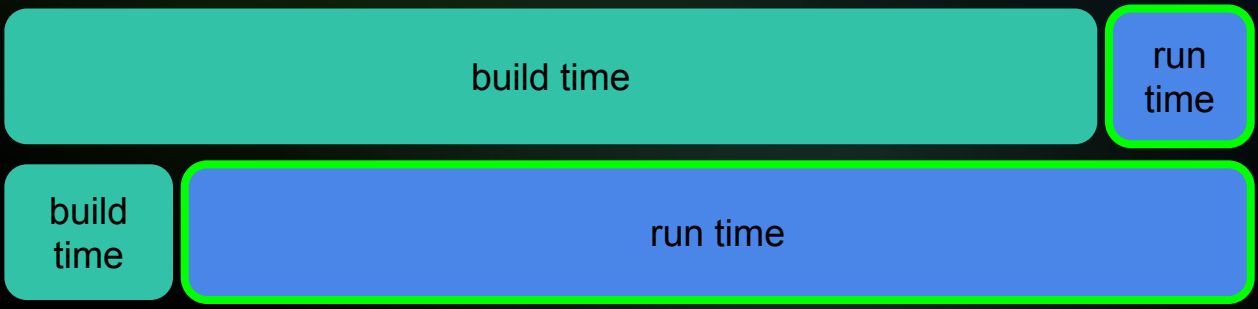
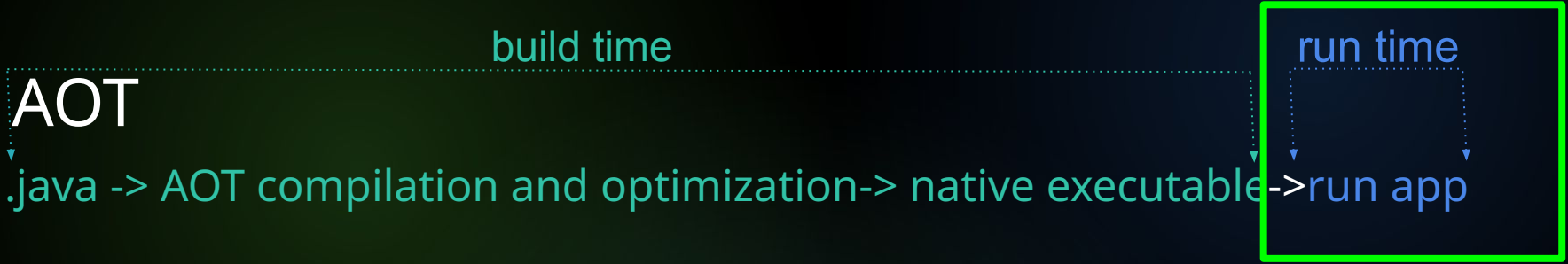
VS

`--initialize-at-run-time`



`--initialize-at-build-time` VS `--initialize-at-run-time`

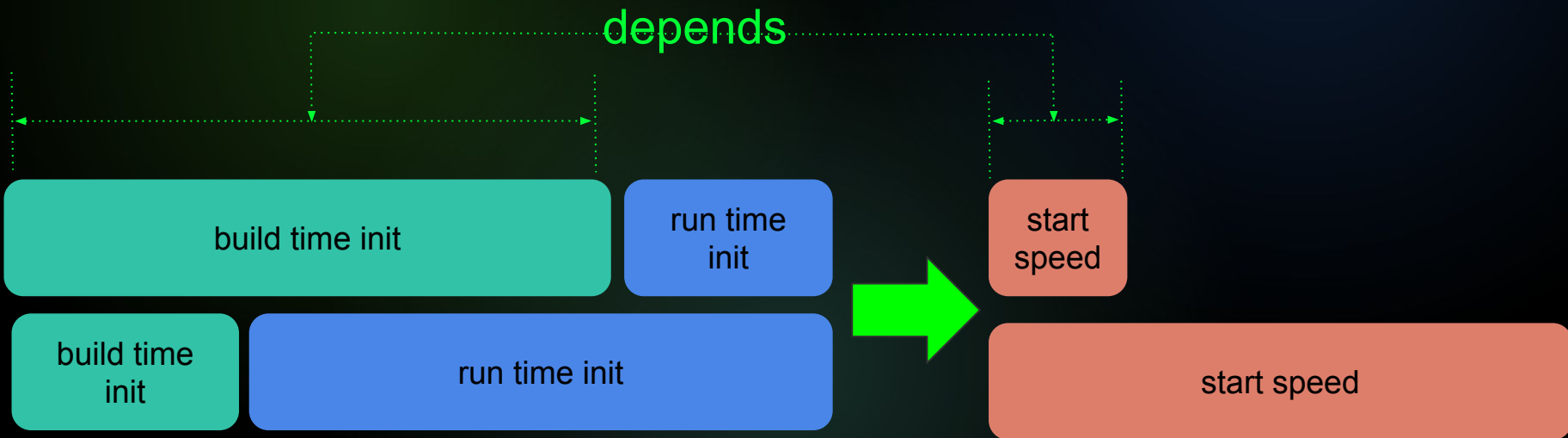
AOT



`--initialize-at-build-time`

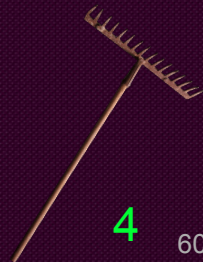
VS

`--initialize-at-run-time`



Пробуем дальше --no-fallback --initialize-at-build-time

```
Терминал Терминал Терминал Терминал Терминал Терминал
#4 ...
#5 [internal] load metadata for docker.io/library/alpine:3.14
#5 DONE 0.5s
#4 [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine
#4 DONE 0.5s
#6 [internal] load .dockerignore
#6 transferring context: 2B done
#6 DONE 0.0s
#7 [stage-3 1/6] FROM docker.io/library/alpine:3.14@sha256:0f2d5c38dd7a4f4f733e688e3a6733cb5ab1ac6e3cb4603a5dd564e5bfb80eed
#7 DONE 0.0s
#8 [buildnative 1/4] FROM ghcr.io/graalvm/native-image-community:17
#8 DONE 0.0s
#9 [m2 1/1] FROM docker.io/library/m2image:latest
#9 CACHED
#10 [buildnative 2/4] WORKDIR /app
#10 CACHED
#11 [stage-3 2/6] WORKDIR /app
#11 CACHED
#12 [build 1/10] FROM docker.io/library/openjdk:17-jdk-alpine@sha256:4b6abae565492dbe9e7a894137c966a7485154238902f2f25e9dbd9784383d81
#12 CACHED
#13 [internal] load build context
#13 transferring context: 62.46kB 0.0s done
#13 DONE 0.0s
#14 [stage-3 3/6] COPY target/client.jks .
#14 DONE 0.1s
#15 [build 2/10] COPY --from=m2 /root/.m2/repository /root/.m2/repository
#15 ...
#16 [stage-3 4/6] COPY target/trust.jks .
#16 DONE 0.1s
#15 [build 2/10] COPY --from=m2 /root/.m2/repository /root/.m2/repository
□
```



native app

Machine Code

Substrate VM

- garbage collector
- thread scheduler
- ...

-Application
-libraries

Heap snapshot

native app

Machine Code

Substrate VM

- garbage collector
- thread scheduler
- ...

-Application
-libraries

Heap snapshot

`static ..`

`static {`

`...`

`}`

Пробуем дальше --no-fallback --initialize-at-build-time

Detailed message:

Trace: Object was reached by

trying to constant fold static field `org.apache.ignite.internal.util.IgniteUtils.RND`

`org.apache.ignite.internal.util.IgniteUtils.randomServerNode(IgniteUtils.java:11376)`

parsing method

`org.apache.ignite.internal.util.IgniteUtils.randomServerNode(IgniteUtils.java:11374)`

reachable via the parsing context at static root method.(Unknown Source)

Пробуем дальше --no-fallback --initialize-at-build-time

```
public abstract class IgniteUtils {
...
/** Random is used to get random server node to authentication from client node. */
private static final Random RND = new Random(System.currentTimeMillis());
}

/**
 * @param ctx Kernel context.
 * @return Random alive server node.
 */
public static ClusterNode randomServerNode(GridKernalContext ctx) {
    Collection<ClusterNode> aliveNodes = ctx.discovery().aliveServerNodes();
    int rndIdx = RND.nextInt(aliveNodes.size()) + 1;
    ...
    return rndNode;
}
}
```


native app

Machine Code

Substrate VM

- garbage collector
- thread scheduler
- ...

-Application
-libraries

Heap snapshot



```
static Random RND = new Random();
```

```
static final KeyPairGenerator KEY_PAIR_GEN;  
static final Cipher CIPHER;  
static {  
    try {  
        KEY_PAIR_GEN = KeyPairGenerator.getInstance("RSA");  
        KEY_PAIR_GEN.initialize(1024);  
        CIPHER = Cipher.getInstance("RSA");  
    } catch (NoSuchAlgorithmException e) {  
        throw new RuntimeException(e);  
    }  
}
```

```
KEY_PAIR_GEN.initialize(1024);
```

```
JCAUtil.getDefSecureRandom()
```

```
private static volatile SecureRandom def= new SecureRandom();
```

#26 19.42 Error: Detected an instance of **Random/SplittableRandom** class in the image heap. Instances created during image generation have cached seed values and don't behave as expected.

at java.security.SecureRandom.<init>(SecureRandom.java:224)

at sun.security.jca.JCAUtil.getDefSecureRandom(JCAUtil.java:96)

at java.security.KeyPairGenerator.initialize(KeyPairGenerator.java:376)

Пробуем дальше `--no-fallback --initialize-at-build-time`

Error: Detected an instance of `Random/SplittableRandom` class in the image heap. Instances created during image generation have cached seed values and don't behave as expected. To see how this object got instantiated use `--trace-object-instantiation=java.util.Random`. The object was probably created by a class initializer and is reachable from a static field. You can request class initialization at image runtime by using the option `--initialize-at-run-time=<class-name>`. Or you can write your own initialization methods and call them explicitly from your main entry point.

Пробуем дальше `--initialize-at-run-time`

```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
WORKDIR /app
COPY --from=build target/joker-ignite-1.1.jar ./app.jar
RUN native-image
--static
--no-fallback
--initialize-at-build-time
--initialize-at-run-time=o.a.ignite.internal.util.IgniteUtils
-jar app.jar
```

...

И так несколько приседаний...

Что в итоге получилось?

```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
WORKDIR /app
COPY --from=build target/joker-ignite-1.1.jar ./app.jar
RUN native-image
--static
--no-fallback
--initialize-at-build-time
--initialize-at-run-time=org.apache.ignite.internal.util.Ignite
Utils\,org.apache.ignite.internal.binary.BinaryContext\,org.apa
che.ignite.internal.util.GridUnsafe\,org.apache.ignite.events.E
ventType\,org.apache.ignite.marshaller.MarshallerExclusions\,or
g.apache.ignite.internal.processors.query.QueryUtils\,org.apach
e.ignite.lang.IgniteUuid\,org.apache.ignite.internal.marshaller
.optimized.OptimizedMarshallerUtils\,org.apache.ignite.marshall
er.jdk.JdkMarshaller
-jar app.jar
```

Пробуем дальше собрать...

```
andrey@andrey-desktop:~/IdeaProjects/joker-ignite/4.5native-fallnack$ sh build.sh
Untagged: native_0 5:2
Deleted: sha256:a37ff11a4bf421ca6949abf732cb9ef40b7f8b9f55e4b1a2991c7d8b65106b9be
#0 building with "default" instance using docker driver

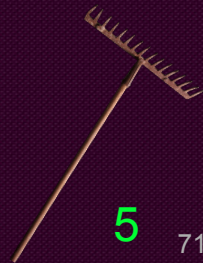
#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 2.59kB done
#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/m2image:latest
#2 DONE 0.0s

#3 [internal] load metadata for ghcr.io/graalvm/native-image-community:17
#3 DONE 0.0s

#4 [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine

```

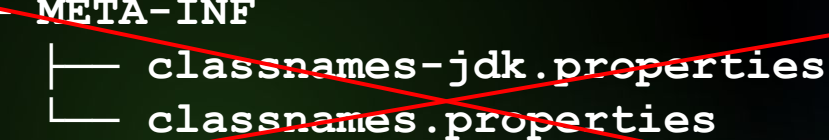


Resource


```
Exception in thread "main" class
org.apache.ignite.IgniteException: Failed to load class
names properties file packaged with ignite binaries
[file=META-INF/classnames.properties]
```



```
|— ignite-core
  |— java
    ...
  |— resources
    |— ignite.properties
    |— META-INF
      |— classnames-jdk.properties
      |— classnames.properties
```



```
public static void processSystemClasses(
ClassLoader classLoader,
@Nullable Collection<PluginProvider> plugins,
Consumer<String> proc) throws IOException {
    Enumeration<URL> urls = classLoader.getResources("META-INF/classnames.properties");
    ...
}
```



resource-config.json

```
{
  "resources": {
    "includes": [
      {
        "pattern": "\\QMETA-INF/classnames-jdk.properties\\E"
      },
      {
        "pattern": "\\QMETA-INF/classnames.properties\\E"
      }
    ]
  },
  "bundles": []
}
```

-H:ResourceConfigurationFiles или
положить в src/main/resources/META-INF/native-image

Пробуем собрать дальше

```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
WORKDIR /app
COPY --from=build target/joker-ignite-1.1.jar ./app.jar
RUN native-image
--static
--no-fallback
--initialize-at-build-time
--initialize-at-run-time=o.a.ignite.internal.util.IgniteUtils...
-H:ResourceConfigurationFiles=../resource-config.json
-jar app.jar
```

Пробуем дальше собрать...

```
andrey@andrey-desktop:~/IdeaProjects/joker-ignite/4.5native-fallnack$ sh build.sh  
Error response from daemon: No such image: native_0_5:2  
#0 building with "default" instance using docker driver
```

```
#1 [internal] load build definition from Dockerfile  
#1 transferring dockerfile: 2.59kB done  
#1 DONE 0.0s
```

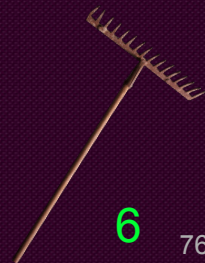
```
#2 [internal] load metadata for ghcr.io/graalvm/native-image-community:17  
#2 DONE 0.0s
```

```
#3 [internal] load metadata for docker.io/library/m2image:latest  
#3 DONE 0.0s
```

```
#4 [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine  
#4 ...
```

```
#5 [internal] load metadata for docker.io/library/alpine:3.14  
#5 DONE 0.5s
```

```
█
```



Reflection

Exception in thread "main" class org.apache.ignite.binary.BinaryObjectException:

Failed to find empty constructor for class: org.apache.ignite.lang.IgniteUuid

at org.apache.ignite.internal.binary.BinaryClassDescriptor.constructor(BinaryClassDescriptor.java:1094)

at org.apache.ignite.internal.binary.BinaryClassDescriptor.<init>(BinaryClassDescriptor.java:309)

at org.apache.ignite.internal.binary.BinaryClassDescriptor.<init>(BinaryClassDescriptor.java:154)

at org.apache.ignite.internal.binary.BinaryContext.registerPredefinedType(BinaryContext.java:1057)

...

reflect-config.json

```
[
  {
    "name": "org.apache.ignite.lang.IgniteUuid",
    "methods": [
      {
        "name": "<init>",
        "parameterTypes": []
      },
      {
        "name": "readResolve",
        "parameterTypes": []
      },
      {
        "name": "writeReplace",
        "parameterTypes": []
      }
    ]
  }
]
```

```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
WORKDIR /app
COPY --from=build target/joker-ignite-1.1.jar ./app.jar
RUN native-image
--static
--no-fallback
--initialize-at-build-time
--initialize-at-run-time=o.a.ignite.internal.util.IgniteUtils...
-H:ResourceConfigurationFiles=./resource-config.json
-H:+PrintAnalysisCallTree -H:PrintAnalysisCallTreeType=CSV
-jar app.jar
```

Пробуем дальше...

resource-config.json +
reflect-config.json +
jni-config.json
predefined-classes-config.json
proxy-config.json
serialization-config.json



```
FROM ghcr.io/graalvm/native-image-community:17 as build-native
WORKDIR /app
COPY --from=build target/joker-ignite-1.1.jar ./app.jar
RUN native-image
--static
--no-fallback
--initialize-at-build-time
--initialize-at-run-time=o.a.ignite.internal.util.IgniteUtils...
-H:ResourceConfigurationFiles=../resource-config.json
-jar app.jar
```

+

```
resource-config.json
reflect-config.json
jni-config.json
predefined-classes-config.json
proxy-config.json
serialization-config.json
```

И нет – опять ничего не вышло.

```
Терминал Терминал Терминал Терминал Терминал Терминал
andrey@andrey-desktop:~/IdeaProjects/joker-ignite/5native-files$ sh build.sh
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 2.04kB done
#1 DONE 0.0s

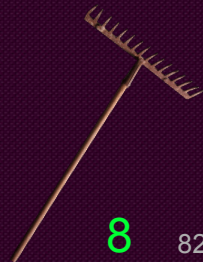
#2 [internal] load metadata for docker.io/library/m2image:latest
#2 DONE 0.0s

#3 [internal] load metadata for ghcr.io/graalvm/native-image-community:17
#3 DONE 0.0s

#4 [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine
#4 DONE 0.5s

#5 [internal] load metadata for docker.io/library/alpine:3.14

```



И нет – опять ничего не вышло.

```
Exception in thread "main" class org.apache.ignite.binary.BinaryObjectException:  
Failed to get constructor for class:  
org.apache.ignite.i.p.platform.websession.PlatformDotNetSessionLockResult at  
org.apache.ignite  
.internal.binary.BinaryClassDescriptor.constructor  
(BinaryClassDescriptor.java:1101)
```



ШО? ОПЯТЬ?

И нет – опять ничего не вышло.

Exception in thread "main" class org.apache.ignite.binary.BinaryObjectException:

Failed to get constructor for class:

org.apache.ignite.i.p.platform.websession.PlatformDotNetSessionLockResult at
org.apache.ignite.internal.binary.BinaryClassDescriptor.constructor
(BinaryClassDescriptor.java:1101)

version 2.14

```
public class PlatformDotNetSessionLockResult ... {  
    private boolean success;  
    ...  
}
```

version > 2.14

```
public class PlatformDotNetSessionLockResult ... {  
    private boolean success;  
    ...  
  
    /**  
     * Default constructor for deserialization.  
     */  
    public PlatformDotNetSessionLockResult () {  
        // No-op.  
    }  
}
```



@Substitute

@Substitute

```
@TargetClass(...PlatformDotNetSessionLockResult.class)
@Substitute
public final class Target_PlatformDotNetSessionLockResult
{
    /**
     * Default constructor for deserialization.
     */
    public Target_PlatformDotNetSessionLockResult() {
        // No-op.
    }
}
```



docker build

docker run --cpus=0.01

```
Терминал Терминал Терминал Терминал Терминал Терминал
andrey@andrey-desktop:~/IdeaProjects/joker-ignite/5native-files$ sh build.sh
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 2.06kB done
#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/m2image:latest
#2 DONE 0.0s

#3 [internal] load metadata for ghcr.io/graalvm/native-image-community:17
#3 DONE 0.0s

#4 [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine
#4 DONE 1.0s

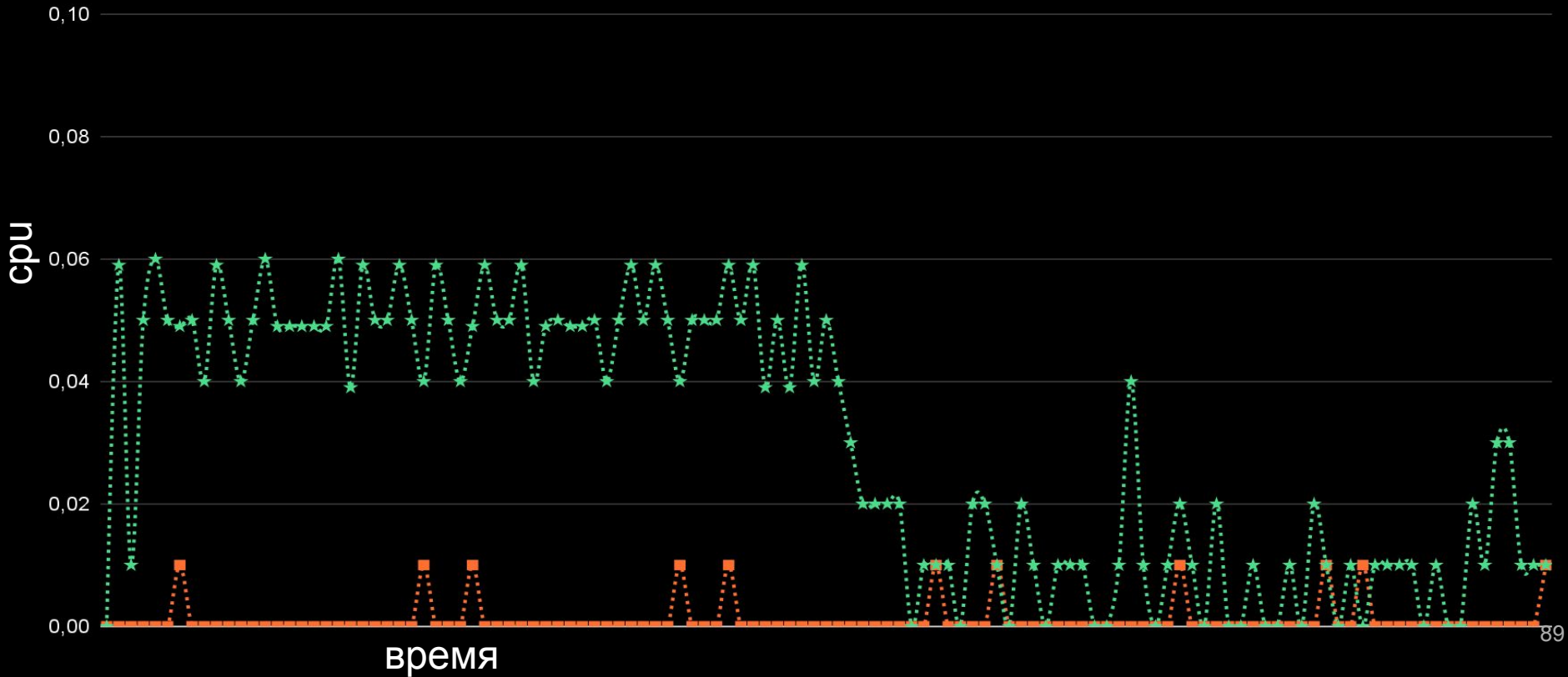
#5 [internal] load metadata for docker.io/library/alpine:3.14
□
```

Что получилось.

спи	Время старта
1	~0.05 s
0.5	~0.08 s
0.3	~0.08 s
0.1	~0.61 s
0.01	~1.90 s

--cpus=0.05

■ native ★ jvm



Real (MB)

■ native ★ jvm

память МВ

100

75

50

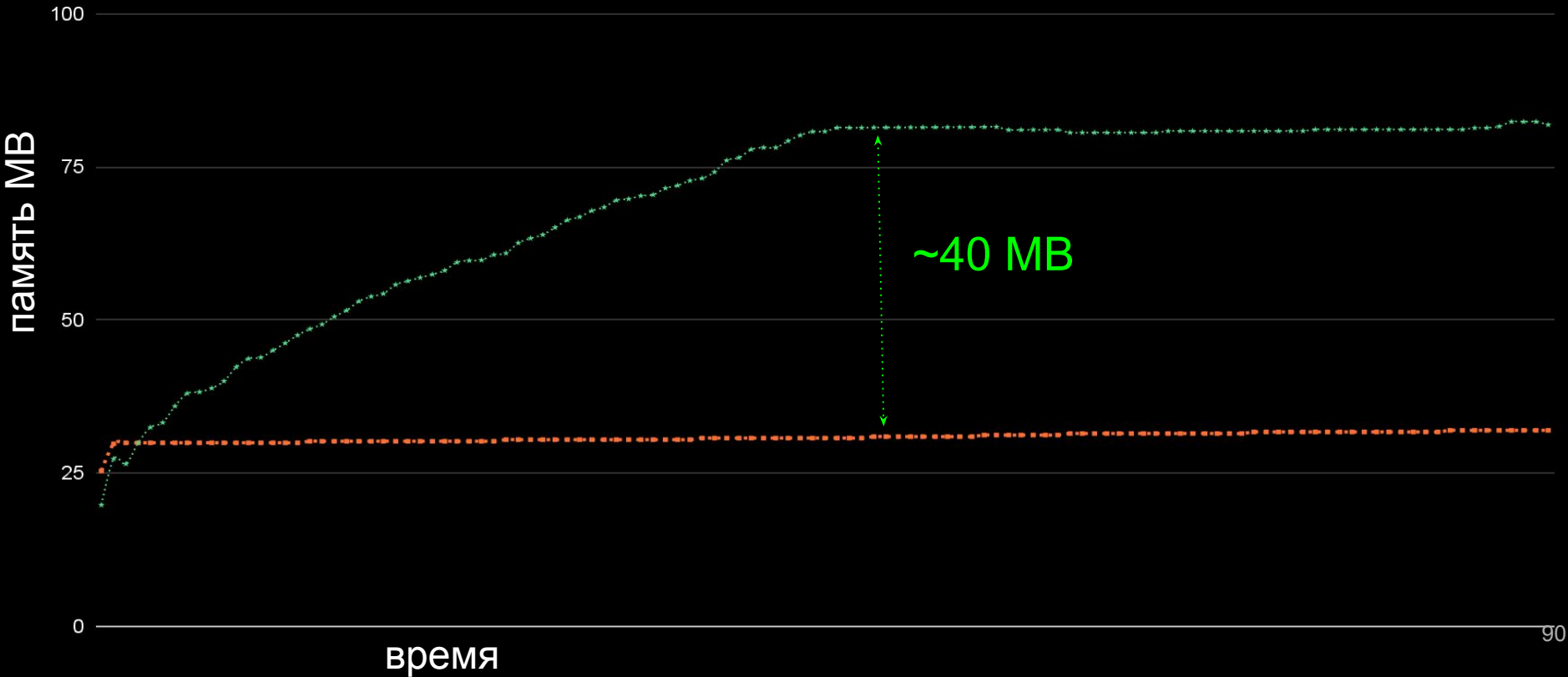
25

0

время

90

~40 MB



Но все равно есть вопросы!

- Как постоянно собирать все эти json-ы?



resource-config.json
reflect-config.json
jni-config.json
predefined-classes-config.json
proxy-config.json
serialization-config.json

Но все равно есть вопросы!

- Как постоянно собирать все эти json-ы?
- Как это тестировать?



JIT

`.java -> javac -> bytecode -> jvm -> tests`

AOT

`.java -> bytecode -> AOT -> tests?`

unit
integration

..

unit 50/50
integration

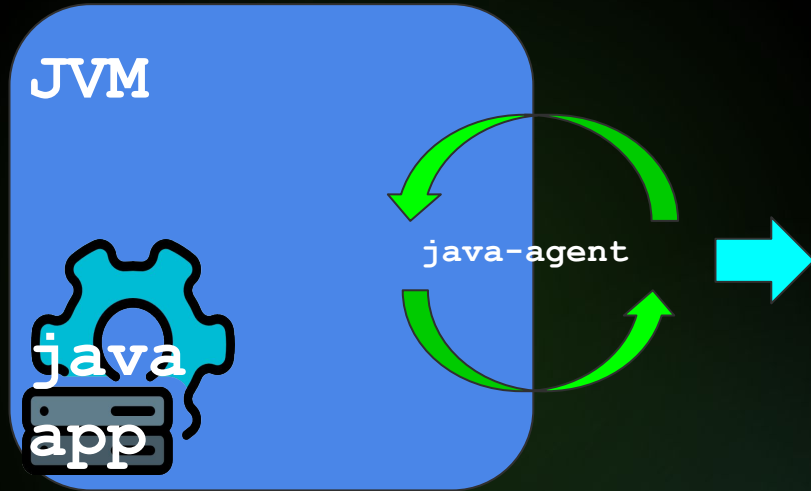
Есть несколько решений

- 1) "Плохое"
- 2) "Хорошее"
- 3) Хорошее, но не всегда возможное
- 4) Очень Хорошее



Все решения имеют
преимущества

“Плохое”: java-agent



```
resource-config.json  
reflect-config.json  
jni-config.json  
predefined-classes-config.json  
proxy-config.json  
serialization-config.json
```

“Плохое”: multi stage build jar

```
FROM native-image-community:17 AS trace
WORKDIR /app
COPY pom.xml .
COPY src ./src
COPY mvnw .
COPY .mvn .mvn
RUN ./mvnw clean test
```

-agentlib:native-image-agent=conf=dir

“Плохое”: multi stage package

```
FROM native-image-community:17 as tracer
```

```
FROM openjdk:17-jdk-alpine AS metha
```

```
WORKDIR /app
```

```
COPY pom.xml .
```

```
COPY src ./src
```

```
COPY --from=tracer configs .resources/META-INF/native-image
```

```
COPY mvnw .
```

```
COPY .mvn .mvn
```

```
RUN ./mvnw clean package
```


“Плохое”: multi stage package

```
FROM native-image-community:17 as tracer
```

```
FROM openjdk:17-jdk-alpine AS metha
```

```
FROM native-image-community:17 as build-native-test
```

```
WORKDIR /app
```

```
COPY --from=metha |joker-ignite-tests1.0.jar| ./app.jar
```

```
RUN native-image joker-ignite-1.1-tests.jar
```

```
--no-fallback
```

```
-o /target/native-tests
```

```
-D.j.p.l.uid.tracking.output.dir=/test-ids
```

```
org.graalvm.junit.platform.NativeImageJUnitLauncher
```

```
RUN ./native-tests
```

“Плохое”: multi stage package

```
FROM native-image-community:17 as tracer
```

```
FROM openjdk:17-jdk-alpine AS metha
```

```
FROM native-image-community:17 as build-native-test
```

```
FROM native-image-community:17 as build-native
```

```
WORKDIR /app
```

```
COPY --from=metha joker-ignite-1.0.jar ./app.jar
```

```
RUN native-image -jar app.jar
```

“Плохое”: multi stage package

```
FROM openjdk:17-jdk-alpine AS build
```

```
FROM native-image-community:17 as tracer
```

```
FROM openjdk:17-jdk-alpine AS metha
```

```
FROM native-image-community:17 as buildnative
```

```
FROM alpine:3.14
```

```
WORKDIR /app
```

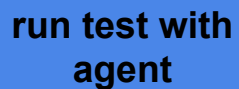
```
COPY --from=buildnative /app /app
```

```
ENTRYPOINT ["/app"]
```

“Хорошее”: native-maven(gradle)-plugin

```
<plugin>
  <groupId>org.graalvm.buildtools</groupId>
  <artifactId>native-maven-plugin</artifactId>
  <version>${native.maven.plugin.version}</version>
  <extensions>true</extensions>
  <executions>
    <execution>
      ...
    </execution>
  </executions>
  <configuration>
    <buildArgs>
      --static --no-fallback --initialize-at-run-time=...--initialize-at-build-time
    </buildArgs>
  </configuration>
</plugin>
```

run test with
agent



```
[INFO]-----  
[INFO]  T E S T S  
[INFO]-----  
[DEBUG]  
Forking command line: /bin/sh -c cd '/app' &&  
'/graalvm-community-java17/java'  
'-agentlib:native-image-agent=config-output-dir=/dir'  
'-jar' '/test.jar'
```

run test with
agent

merge
meta

```
[INFO] Results: Tests run: 5, Errors: 0
```

```
[DEBUG] org.graalvm.buildtools:native-maven-plugin:0.10.3: merge-agent-files
```

```
[DEBUG] Agent output:
```

```
jni-config.json,  
predefined-classes-config.json,  
serialization-config.json,  
resource-config.json,  
proxy-config.json,  
reflect-config.json
```

run test with agent

merge meta

build-test image

```
[INFO] /native-tests
-Djunit.p.l.uid.tracking.output.dir=/test-ids
//найдет
--features=p.JUnitPlatformFeature
//запустит
o.g.j.p.NativeImageJUnitLauncher
```

Top 10 origins of code area:

11.44MB java.base

3.79MB java.xml

1.50MB ignite-core-2.14.0.jar

1.11MB svm.jar (Native Image)

50.99kB java.rmi

337.95kB jdk.crypto.ec

206.14kB junit-jupiter-engine-5.10.0.jar

204.37kB java.naming 541.03kB

151.97kB junit-platform-launcher-1.10.0.jar

133.09kB jdk.naming.dns

950.80kB for 25 more packages

run test with agent

merge meta

build-test image

run native test

```
ru.joker.test.IgniteTest > createCacheTest() SUCCESSFUL  
ru.joker.test.IgniteTest > getAndPutCache() SUCCESSFUL  
ru.joker.test.IgniteTest > testFunctional() SUCCESSFUL  
ru.joker.test.IgniteTest > getANdPutBinaryCache() SUCCESSFUL  
ru.joker.test.IgniteTest > testAllMethodCache() SUCCESSFUL
```

Test run finished after 108 ms

```
[ 2 containers found ]  
[ 0 containers skipped ]  
[ 2 containers started ]  
[ 0 containers aborted ]  
[ 2 containers successful ]  
[ 0 containers failed ]  
[ 5 tests found ]  
[ 0 tests skipped ]  
[ 5 tests started ]  
[ 0 tests aborted ]  
[ 5 tests successful ]  
[ 0 tests failed ]
```


run test with agent

merge meta

build-test image

run native test

build image

```
[DEBUG] native-maven-plugin:build:build-native
[1/8] Initializing...
Java version: 17.0.9+9, vendor version: GraalVM CE 17.0.9+9.1
Graal compiler: optimization level:2, target machine:
x86-64-v3
C compiler: gcc (redhat, x86_64, 11.3.1)
Garbage collector: Serial GC (max heap size:80% of RAM)
[2/8] Performing analysis... [*****]
    6,657 (82.46%) of 8,073 types reachable
    9,680 (56.96%) of 16,995 fields reachable
   32,386 (54.47%) of 59,453 methods reachable
    2,169 types,    278 fields, and 1,898 methods registered for
reflection
           64 types,    76 fields, and    55 methods registered for
JNI access
           4 native libraries: dl, pthread, rt, z
[3/8] Building universe...
[4/8] Parsing methods...    [*]
[5/8] Inlining methods...  [***]
[6/8] Compiling methods... [****]
[7/8] Layouting methods... [**]
[8/8] Creating image...    [**]
```

“Хорошее” решение: немного о проблемах

```
<ignite.jvm.options>
```

```
--add-opens=java.base/jdk.internal.access=ALL-UNNAMED
```

```
--add-opens=java.base/jdk.internal.misc=ALL-UNNAMED
```

```
--add-opens=java.base/sun.nio.ch=ALL-UNNAMED
```

```
... .
```

```
</ignite.jvm.options>
```



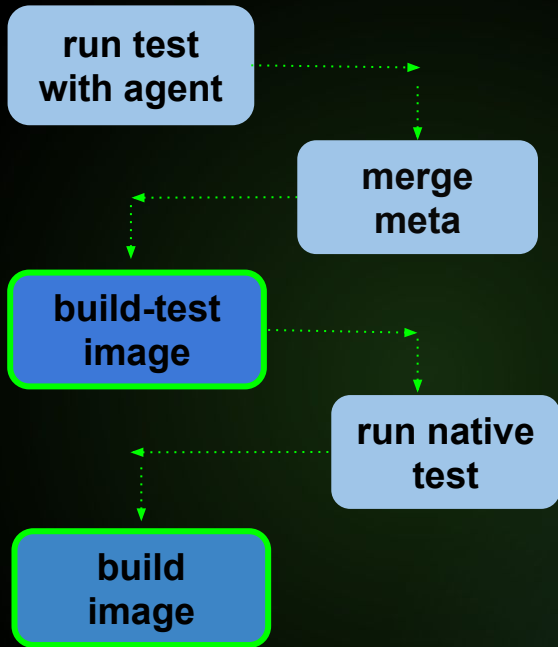
<https://github.com/graalvm/native-build-tools/pull/615/commits/f3d0bf3d38993dde1324f7f8c482f3fa15c575a5>

Хорошее, но не всегда ВОЗМОЖНОЕ

ignite-core

JAR

```
|— ignite-core
  |— java
    ...
  |— resources
    |— ignite.properties
    |— META-INF
      |— classnames-jdk.properties
      |— classnames.properties
      |— native-image
        |— jni-config.json
        |— predefined-classes-config.json
        |— proxy-config.json
        |— reflect-config.json
        |— resource-config.json
        |— serialization-config.json
```



```
Apply jar:file:/.m2/org/apache/ignite/ignite-core/2.15.0/ignite-core-2.15.0.jar!/META-INF/native-image/jni-config.json
```

```
Apply jar:file:/.m2/org/apache/ignite/ignite-core/2.15.0/ignite-core-2.15.0.jar!/META-INF/native-image/reflect-config.json
```

```
Apply jar:file:/.m2/org/apache/ignite/ignite-core/2.15.0/ignite-core-2.15.0.jar!/META-INF/native-image/resource-config.json
```

```
Apply jar:file:/.m2/org/apache/ignite/ignite-core/2.15.0/ignite-core-2.15.0.jar!/META-INF/native-image/proxy-config.json
```

...

Очень Хорошее: graalvm-reachability-metadata

```
<plugin>
  <groupId>org.graalvm.buildtools</groupId>
  <artifactId>native-maven-plugin</artifactId>
  <version>${native.maven.plugin.version}</version>
  <extensions>true</extensions>
  <executions>
    <execution>
      ...
    </execution>
  </executions>
  <configuration>
    <buildArgs>
      --static --no-fallback --initialize-at-run-time=...--initialize-at-build-time
    </buildArgs>
    <metadataRepository>
      <enabled>true</enabled>
      <version>0.2.3</version>
    </metadataRepository>
  </configuration>
</plugin>
```

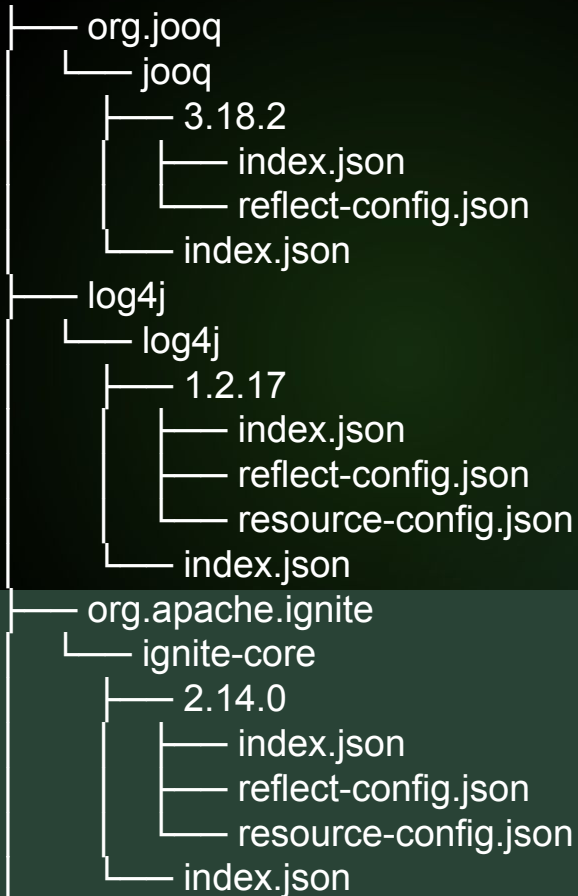
Очень Хорошее: graalvm-reachability-metadata

ch.qos.logback
ch.qos.logback.contrib
com.ecwid.consul
com.fasterxml.jackson.core
com.github.ben-manes.caffeine
com.github.luben
com.google.protobuf
com.graphql-java
com.h2database
com.hazelcast
com.itextpdf
com.microsoft.sqlserver
com.mysql
com.sun.mail
com.zaxxer
commons-logging
io.grpc
io.jsonwebtoken
io.nats
io.netty
io.opentelemetry
io.undertow
jakarta.servlet
javax.cache
log4j
mysql

org.apache.activemq
org.apache.commons
org.apache.httpcomponents
org.apache.kafka
org.apache.tomcat
org.apache.tomcat.embed
org.bouncycastle
org.eclipse.angus
org.eclipse.jetty
org.eclipse.jgit
org.eclipse.paho
org.ehcache
org.example
org.flywaydb
org.freemarker
org.glassfish.jaxb
org.hdrhistogram
org.hibernate
org.hibernate.orm
org.hibernate.reactive
org.hibernate.validator
org.jboss.logging
org.jboss.spec.javaax.servlet
org.jctools
org.jetbrains.kotlin
org.jline
org.jooq

org.apache.commons
org.apache.httpcomponents
org.apache.kafka
org.apache.tomcat
org.apache.tomcat.embed
org.bouncycastle
org.eclipse.angus
org.eclipse.jetty
org.eclipse.jgit
org.eclipse.paho
org.ehcache
org.example
org.flywaydb
org.freemarker
org.glassfish.jaxb
org.hdrhistogram
org.hibernate
org.hibernate.orm
org.hibernate.reactive
org.hibernate.validator
org.jboss.logging
org.jboss.spec.javaax.servlet
org.jctools
org.jetbrains.kotlin
org.jline
org.jooq
ваша.любимая.библиотека

Очень Хорошее: graalvm-reachability-metadata



```
<metadataRepository>
```

```
<enabled>true</enabled>
```

```
<url>${metadata.url}</url>
```

```
</metadataRepository>
```

Итоги

- Стоит взвесить все за и против
- Стоит понимать, что `native images` применимы для определенного класса приложений
- Рассмотрели часть проблем, с которыми можно столкнуться, и как их решить
- Рассмотрели варианты, с помощью которых можно настроить инфраструктуру сборок так, чтобы все было ок, собиралось и тестировалось
- Получили способ писать легковесные сайдкары для приложений бонусом