CI/CD в большом on-premise Datalake-проекте

Никита Благодарный
telegram @nblagodarnyy

Александра Чекмарева (Китченко)
telegram @sasha_kitchen

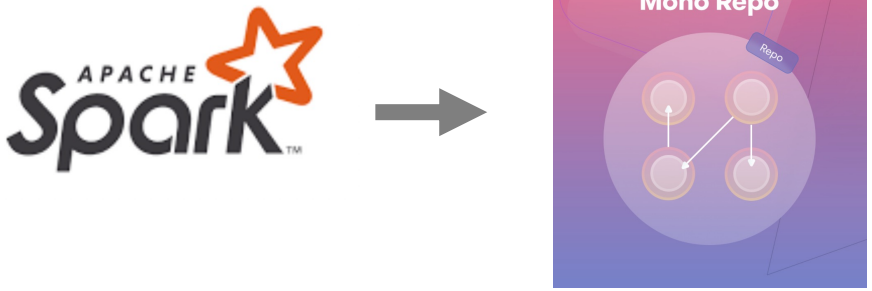
ЧЕСТНЫЙ ЗНАК | центр развития перспективных технологий

SmartData
2024

Немного вводных

Про что этот доклад?

Как структурировать и делать CI/CD

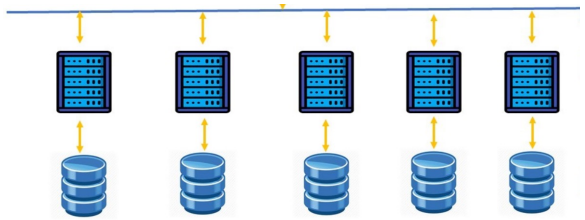


Statistic | Statistic

Refresh | Refresh on selection | Settings

Overview | properties | py | txt | xml

Extension	Lines CODE	Count
sql (*SQL files*)	150268	2804x
scala (*SCALA files*)	98141	1627x
yaml (*YAML files*)	79769	2225x
py (*PY files*)	37789	427x
ddl (*DDL files*)	18012	94x



О себе

Data Engineer, 17 лет в IT, из них 14 в области BI/DE

- 2007-2011 Сбербанк, разработка OLTP-решений на Oracle, ХД/Reporting на стеке Microsoft
- 2012-2016 Сбербанк, разработка ЦХД на Oracle / Teradata / Informatica PC
- 2016-2018 КРОК, проекты на Oracle / Informatica / Hadoop (HDP, Arenadata)/Spark/Hive/NiFi
- 2018-2019 Газпромнефть, построение DataLake (Arenadata, MS SQL, Informatica, Spark/Hive)
- 2019-.... ЦРПТ, DataLake (Vanilla Hadoop/Spark/Hive/HBase/ClickHouse/Postgres/etc)

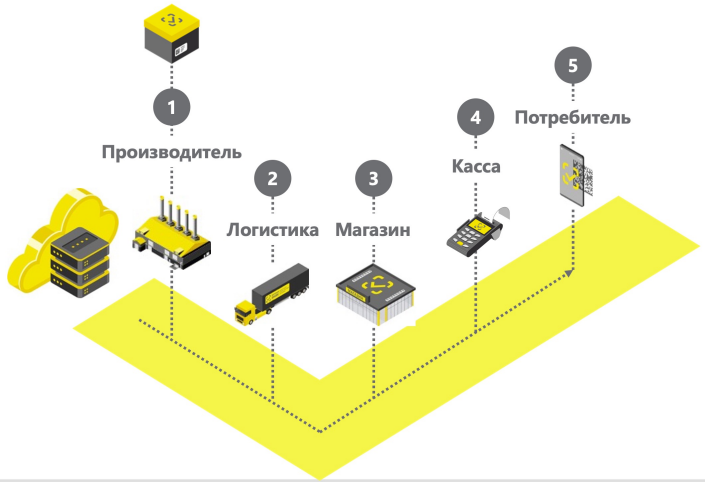
DevOps Engineer, 10 лет в IT, последние 6 лет работаю DevOps/MLOps на проектах Big Data

- 2018-2021 TELE2, DevOps, разработка решений машинного обучения (CI/CD, gitlab, Jenkins, airflow, kubernetes, python, sql, pyspark, scala, hive, sqoop, bteq, Hadoop, Teradata)
- 2021-2023 ЦРПТ (Честный Знак), DevOps Аналитической платформы (ansible, gitlab ci, kubernetes, helm, bash, python, maven, Hadoop, ClickHouse, Postgres, HAproxy)
- 2023-... НЛМК, MLOps, разработка DSML-платформы (CI/CD, gitlab, Minio, S3, airflow, kubernetes, helm, python, Hadoop, Computer Vision, Machine learning)

О компании Честный Знак

Создаем систему цифровой маркировки и прослеживания товаров "Честный Знак" в России и странах ЕАЭС

- Основана в 2017 году
- > 750 сотрудников
- Офисы разработки в Москве, Питере, Пензе, Ульяновске, Нижнем Новгороде
- Разнообразные товарные группы (Табак, Молоко, Вода, Пиво, Лекарства, Одежда, Обувь, ...)



О технологическом ландшафте

Apache Phoenix
Apache HBase
cassandra

4 кластера (180 машин)
~5 Pb хранения

Apache Hadoop
Apache Spark
Hive

Кластер 410 машин
146 Tb RAM
9,5 Pb хранения

ClickHouse

Кластер 74 машины
800 Tb хранения

PostgreSQL
MDM Master Data Management

2 кластера Patroni
~2 Tb хранения

Apache kafka

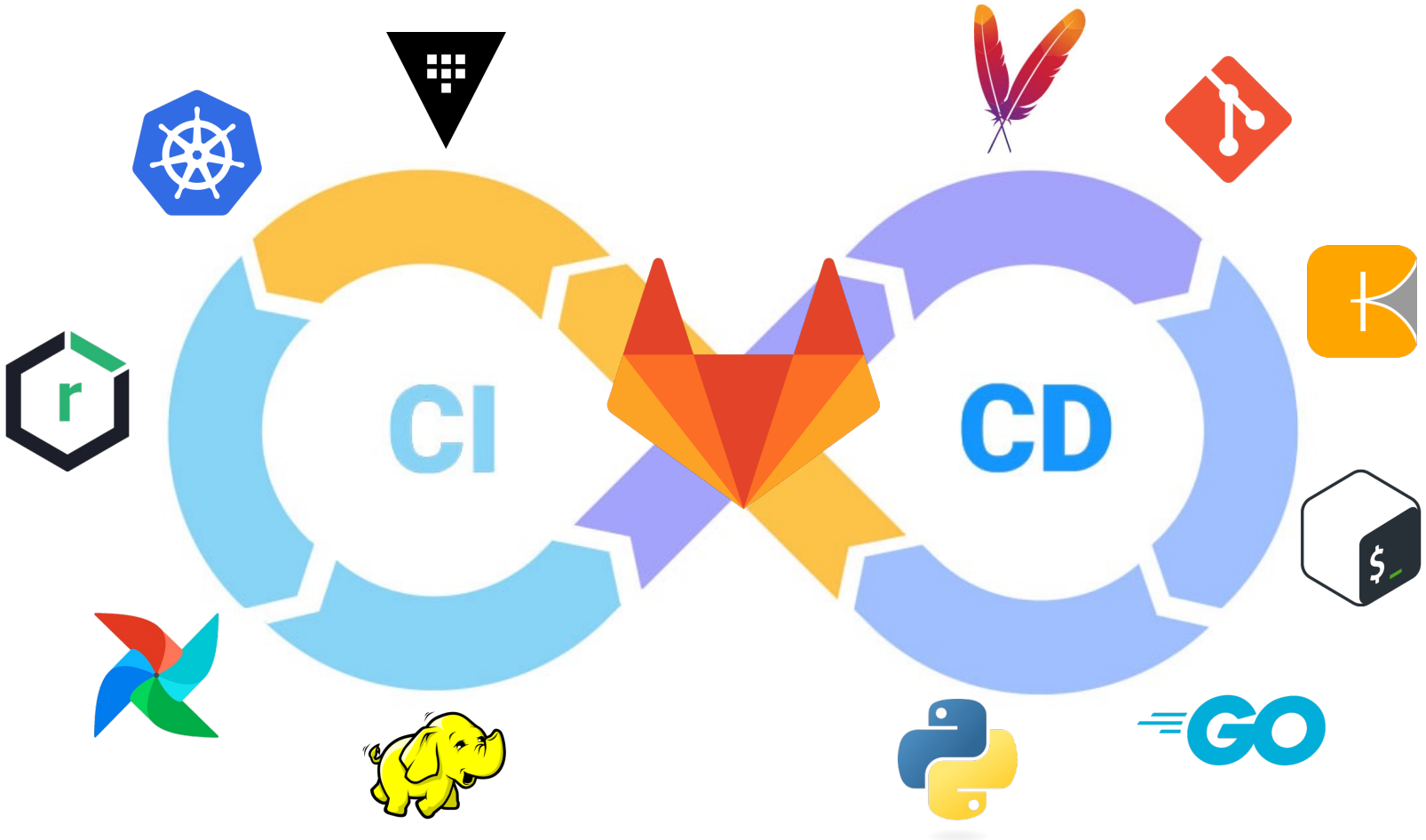
Кластер 18 машин

JasperReports
re:dash
Apache Airflow
Apache Zeppelin
LIVY
jupyterlab
kubernetes
HUE

MINIO

Scala

О технологическом стеке CI/CD

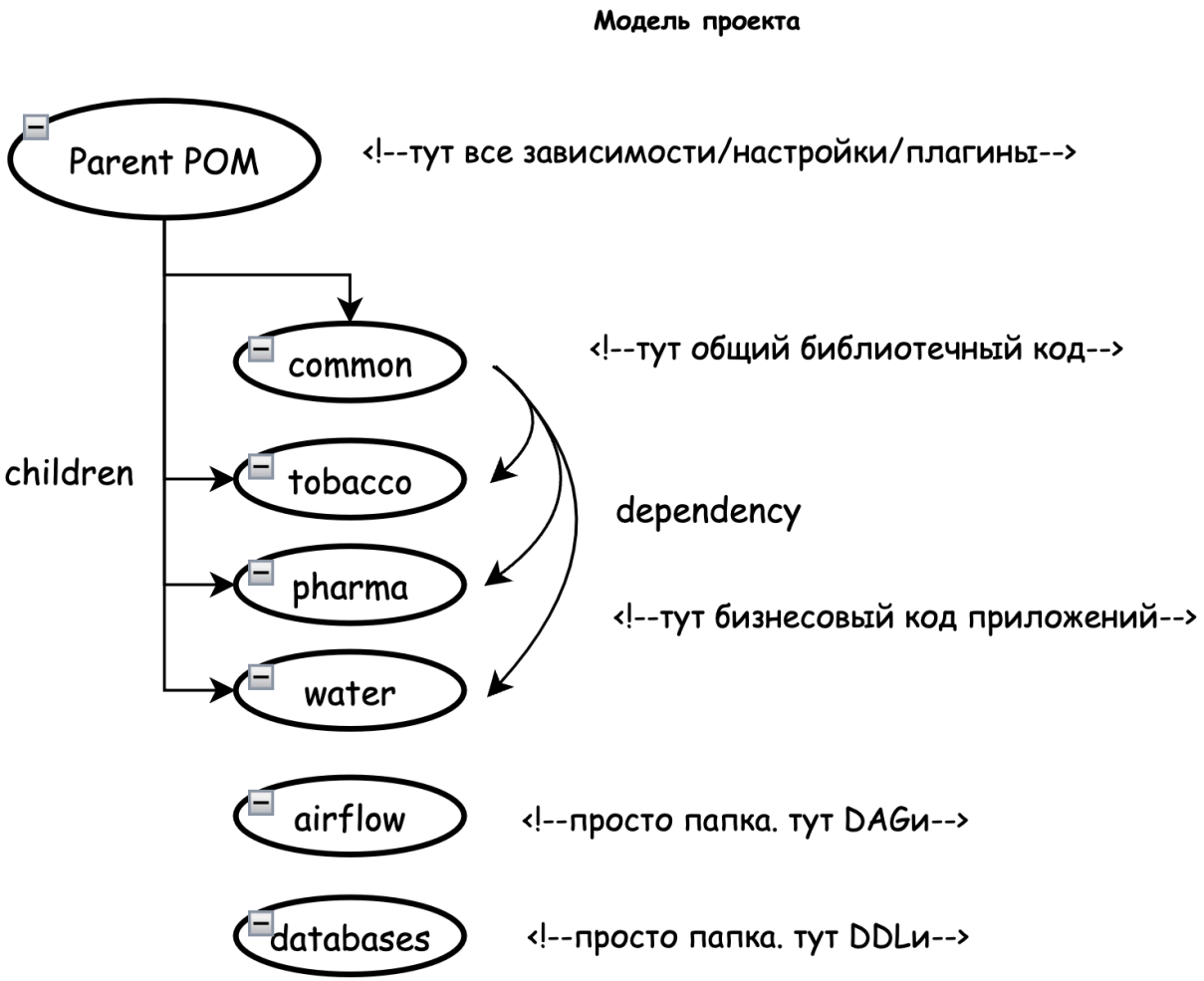


Структура проекта в Git

Структура проекта

Структура проекта – иерархия/зависимости

Maven™




```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http:
    <modelVersion>4.0.0</modelVersion>

    <groupId>ru.crpt.analytics</groupId>
    <artifactId>datalake_etl</artifactId>
    <packaging>pom</packaging>
    <version>${revision}</version>
    <modules>
        <module>common</module>
        <module>clean-data</module>
        <module>tobacco</module>
        <module>pharma</module>
        <module>service_providers</module>
        <module>legprom</module>
        <module>milk</module>
        <module>water</module>
        <module>furs</module>
        <module>master-data</module>
    </modules>
```

Структура проекта – parent POM

Maven™

pom.xml (datalake_etl)

```
2    <project xmlns="http://maven.apache.org/POM/4.0.0"
81       <properties>
95           <!--Scala-->
96           <scala.version.short>2.12</scala.version.short>
97           <scala.version>2.12.15</scala.version>
98           <scala.compile.target.jvm>jvm-1.17</scala.compile.target.jvm>
99           <scalatest.version>3.2.18</scalatest.version>
100          <scalatest.maven.plugin.version>2.2.0</scalatest.maven.plugin.version>
101          <scoverage.plugin.version>2.0.4</scoverage.plugin.version>
102          <scoverage.scalacPluginVersion>2.0.11</scoverage.scalacPluginVersion>
103          <scoverage.aggregate>true</scoverage.aggregate>
104          <scalastyle.maven.plugin.version>1.0.0</scalastyle.maven.plugin.version>
105
```

```
% mvn clean compile -Dscala.version="2.12.14"
```

Структура проекта – parent POM

Maven™

```xml
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-core_${scala.version.short}</artifactId>
            <version>${spark.version}</version>
            <type>test-jar</type>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-catalyst_${scala.version.short}</artifactId>
            <version>${spark.version}</version>
            <type>test-jar</type>
        </dependency>
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-sql_${scala.version.short}</artifactId>
            <version>${spark.version}</version>
            <type>test-jar</type>
        </dependency>

        <dependency>
            <groupId>com.github.mrpowers</groupId>
            <artifactId>spark-fast-tests_${scala.version.short}</artifactId>
            <version>${com.github.mrpowers.version}</version>
        </dependency>
```

- Централизация версий зависимостей
- Явное указание версии для транзитивных зависимостей

A second, and very important use of the dependency management section is to control the versions of artifacts used in transitive dependencies. As an example consider these projects:

Project A:

```xml
1. <project>
2.  <modelVersion>4.0.0</modelVersion>
```

Структура проекта – parent POM

Maven™

pom.xml (datalake_etl)

```
2   <project xmlns="http://maven.apache.org/POM/4.0.0"
450     <build>
478         <plugins>
556             <!-- enable scalatest -->
557             <plugin>
558                 <groupId>org.scalatest</groupId>
559                 <artifactId>scalatest-maven-plugin</artifactId>
560                 <version>${scalatest.maven.plugin.version}</version>
561                 <configuration>
562                     <reportsDirectory>${project.build.directory}/surefire-reports</reportsDirectory>
563                     <junitxml>.</junitxml>
564                     <filereports>TestSuite.txt</filereports>
565                     <parallel>false
```

pom.xml (datalake_etl)

```
2   <project xmlns="http://maven.apache.org/POM/4.0.0"
704     <repositories>
712         <repository>
713             <id>java-dev-team</id>
714             <name>crpt-nexus</name>
715             <url>${nexus.repo.url}/repository/java-dev-team/</url>
716             <layout>default</layout>
717         </repository>
718
719     </repositories>
720
721     <distributionManagement>
722
723         <!--Release repo-->
724         <repository>
725             <uniqueVersion>false</uniqueVersion>
726             <id>crpt-nexus-release</id>
727             <name>Common nexus release repository</name>
728             <!--suppress UnresolvedMavenProperty -->
729             <url>${nexus.repo.url}/repository/analytics-team-release/</url>
730         </repository>
731
732         <!--Snapshot repo-->
733         <snapshotRepository>
734             <uniqueVersion>false</uniqueVersion>
735             <id>crpt-nexus-snapshots</id>
736             <name>Common nexus snapshot repository</name>
737             <!--suppress UnresolvedMavenProperty -->
738             <url>${nexus.repo.url}/repository/java-dev-team-snapshot</url>
739         </snapshotRepository>
740         <downloadUrl>${nexus.repo.url}/repository/java-dev-team-snaphot</downloadUrl>
```

Структура проекта – child POM

Maven™

pom.xml (datalake-common-test)

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    <parent>
        <groupId>ru.crpt.analytics</groupId>
        <version>${revision}</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <properties>
        <testcontainers.scala.version>0.40.9</testcontainers.scala.version>
        <kafka.client.version>2.7.0</kafka.client.version>
        <phoenix.thirdparty.version>1.1.0</phoenix.thirdparty.version>
        <com.github.mrpowers.version>0.23.0</com.github.mrpowers.version>
        <assembly.skipAssembly>true</assembly.skipAssembly>
    </properties>
```

pom.xml (datalake-gismt)

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <build>
        <plugins>
            <plugin>
                <groupId>org.codehaus.mojo</groupId>
                <artifactId>build-helper-maven-plugin</artifactId>
                <version>3.5.0</version>
                <executions>
                    <execution>
                        <id>add-source</id>
                        <phase>generate-sources</phase>
                        <goals>
                            <goal>add-source</goal>
                        </goals>
```

Структура проекта – CI-friendly

Maven™

pom.xml (datalake_etl)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http:
    <modelVersion>4.0.0</modelVersion>

    <groupId>ru.crpt.analytics</groupId>
    <artifactId>datalake_etl</artifactId>
    <packaging>pom</packaging>
    <!--Don't change it -->
    <version>${revision}</version>   Blagodarnyi, 18.12.2020, 15:08
    <modules>
        <module>common</module>
        <module>clean-data</module>
        <module>tobacco</module>
        <module>pharma</module>
```

Starting with Maven 3.5.0-beta-1 you can use the `${revision}`, `${sha1}` and/or `${changelist}` as placeholders for the version in your pom file.

```
mvn -Drevision=1.0.0-SNAPSHOT clean package
```

Разные версии Spark

Структура проекта – разные версии Spark

Overview Active Stale **All**

master

`master` default protected

`master_spark_3.5`

Структура проекта – разные версии Spark

pom.xml (datalake_etl)

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
    <profiles>
        <profile>
            <id>spark-3.3</id>
            <activation>
                <activeByDefault>true</activeByDefault>
            </activation>
            <properties...>
        </profile>

        <profile>
            <id>spark-3.5</id>
            <activation>
                <activeByDefault>false</activeByDefault>
            </activation>
            <properties>
                <!--Spark-->
                <spark.version>3.5.1</spark.version>
                <spark.short.version>3.5</spark.short.version>
                <!--Spark catalyst for compile different versions-->
                <org.codehaus.janino.version>3.1.9</org.codehaus.janino.version>
                <spark.xml.version>0.17.0</spark.xml.version>
                <!--Enforced versions for Spark-->
                <fasterxml.jackson.version>2.15.2</fasterxml.jackson.version>
                <fasterxml.jackson-module-scala.version>2.15.2</fasterxml.jackson-module-scala.version>
                <fasterxml.jackson.databind.version>2.15.2</fasterxml.jackson.databind.version>
                <fasterxml.jackson.yaml.version>2.15.2</fasterxml.jackson.yaml.version>
                <fasterxml.jackson.avro.version>2.15.2</fasterxml.jackson.avro.version>
                <revision>1.0.0_${spark.version}</revision>
                <specific.sources>3_5_0</specific.sources>
                <hadoop.version>3.4.0</hadoop.version>
                <iceberg.version>1.5.2</iceberg.version>
            </properties>
            <dependencies>
                <dependency>
                    <groupId>org.apache.spark</groupId>
                    <artifactId>spark-protobuf_${scala.version.short}</artifactId>
                    <version>${spark.version}</version>
                </dependency>
            </dependencies>
        </profile>
</profiles>   Никита, 27.02.2024, 14:02 • BUILD_SPARK_35
```

```
mvn clean package -P spark-3.3
```

```
mvn clean package -P spark-3.5
```

Структура проекта – разные версии Spark

ProtoParser.scala

```
1  package ru.crpt.analytics.datalake.base
2
3  import org.apache.spark.sql.protobuf.functions.from_protobuf
4
5  object ProtoParser {  new *
6
7  }   You, Moments ago • Uncommitted changes
8
```

Cannot resolve symbol protobuf

Create case class 'protobuf' ⌥⇧↵ More actions... ⌥↵

Структура проекта – разные версии Spark

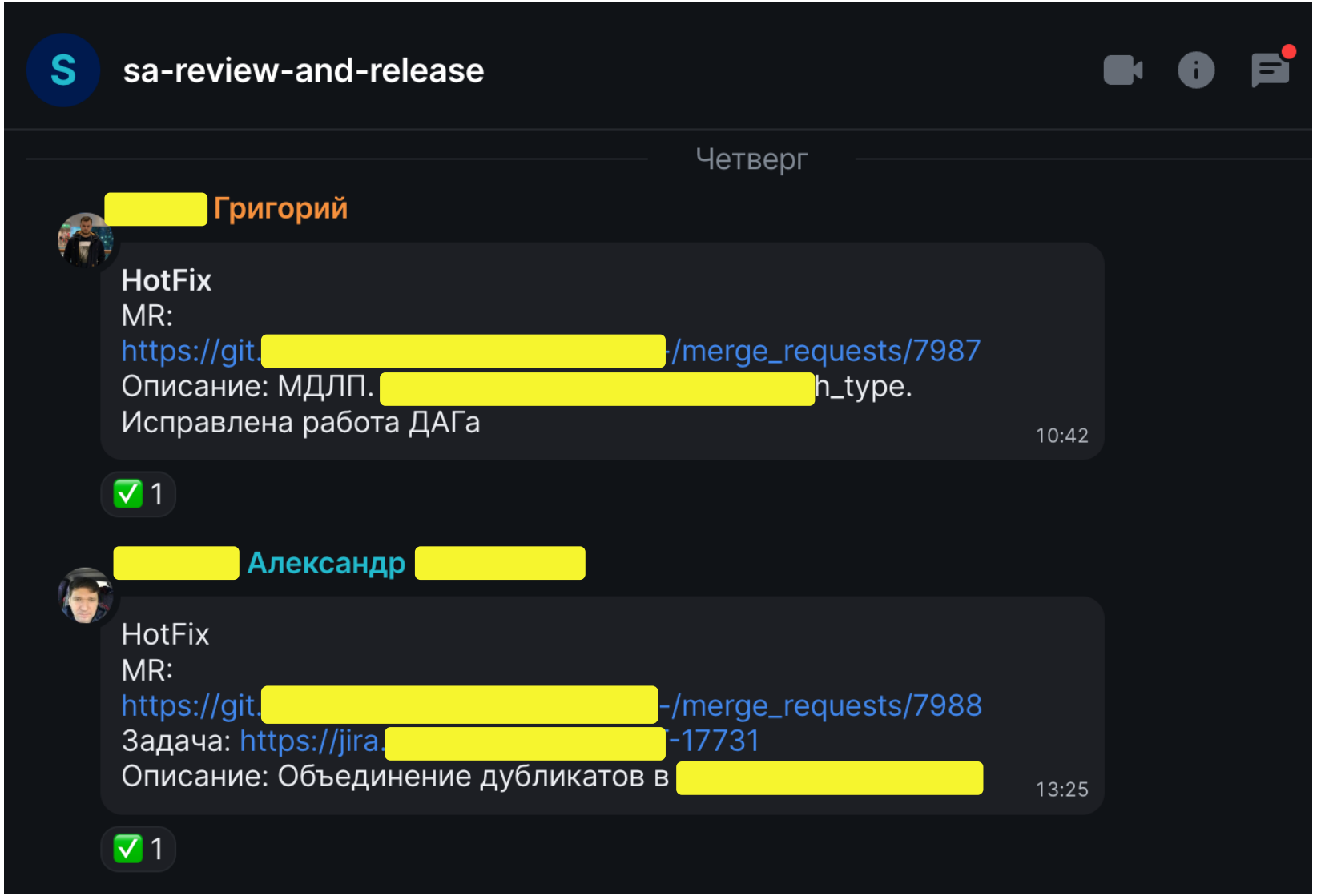
```
hooks
  spark_33 [datalake-spark-hook-3.3]
    src
      main
        resources
        scala
          ru.crpt.analytics.spark_hook
            VersionSpecificTransform
      test
    pom.xml
  spark_35 [datalake-spark-hook-3.5]
    src
      main
        resources
        scala
          ru.crpt.analytics.spark_hook
            VersionSpecificTransform
      test
    pom.xml
```

```
gismt [datalake-gismt]
  src
    main
      resources
      scala
      spark_specific_3_3_0
        ru.crpt.analytics.datalake.gismt
          protoparser
          VersionSpecificTransform
      spark_specific_3_5_0
        ru
          crpt
            analytics
              datalake
                gismt
                  protoparser
                  VersionSpecificTransform
    test
  pom.xml
```

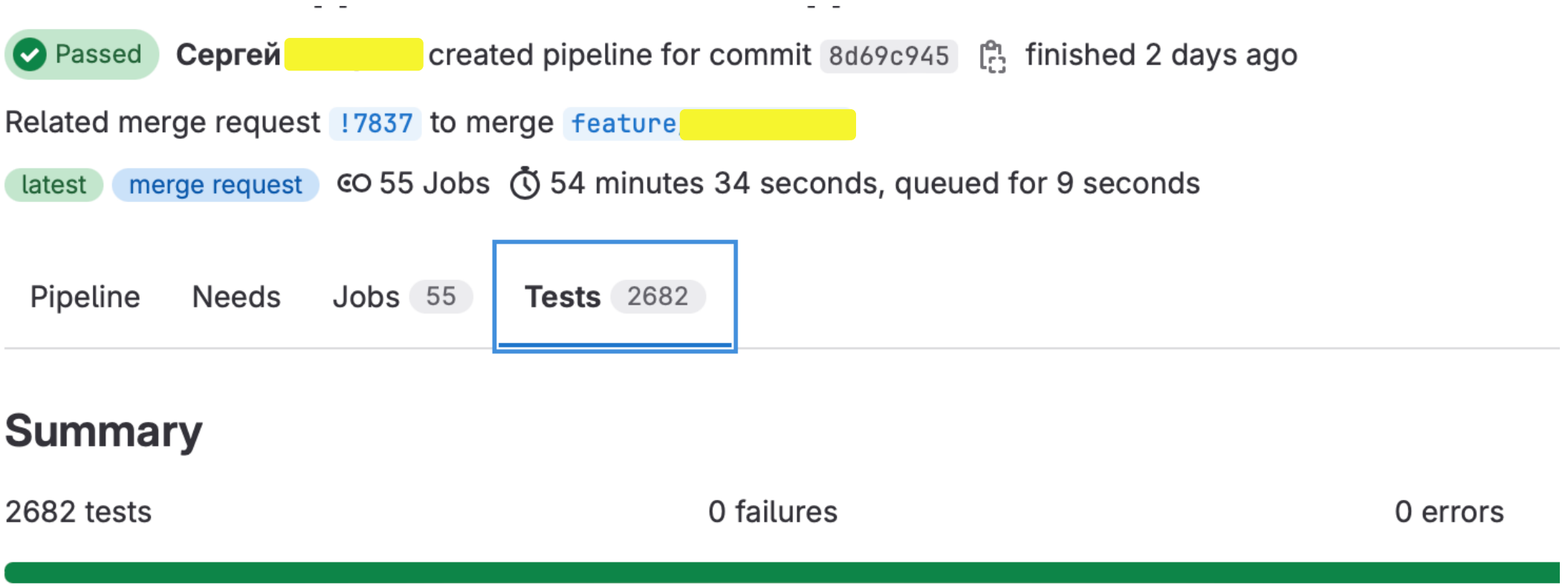
```xml
<plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>build-helper-maven-plugin</artifactId>
    <version>3.5.0</version>
    <executions>
        <execution>
            <id>add-source</id>
            <phase>generate-sources</phase>
            <goals>
                <goal>add-source</goal>
            </goals>
            <configuration>
                <sources>
                    <source>src/main/spark_specific_${specific.sources}</source>
                </sources>
            </configuration>
        </execution>
        <execution>
            <id>add-test-source</id>
            <phase>generate-test-sources</phase>
            <goals>
                <goal>add-test-source</goal>
            </goals>
            <configuration>
                <sources>
                    <source>src/test/spark_specific_${specific.sources}</source>
                </sources>
            </configuration>
        </execution>
    </executions>
</plugin>
```

Процессы в команде

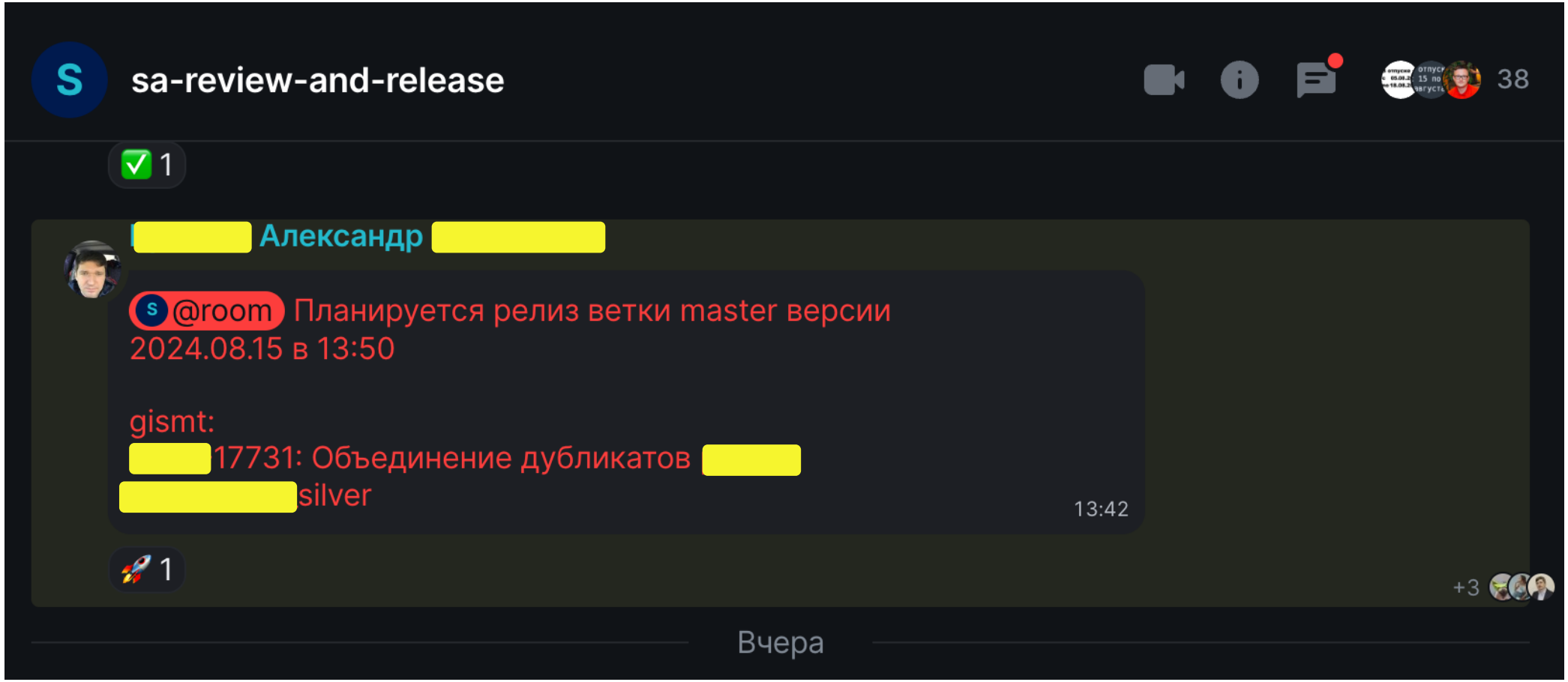
Процессы – code review



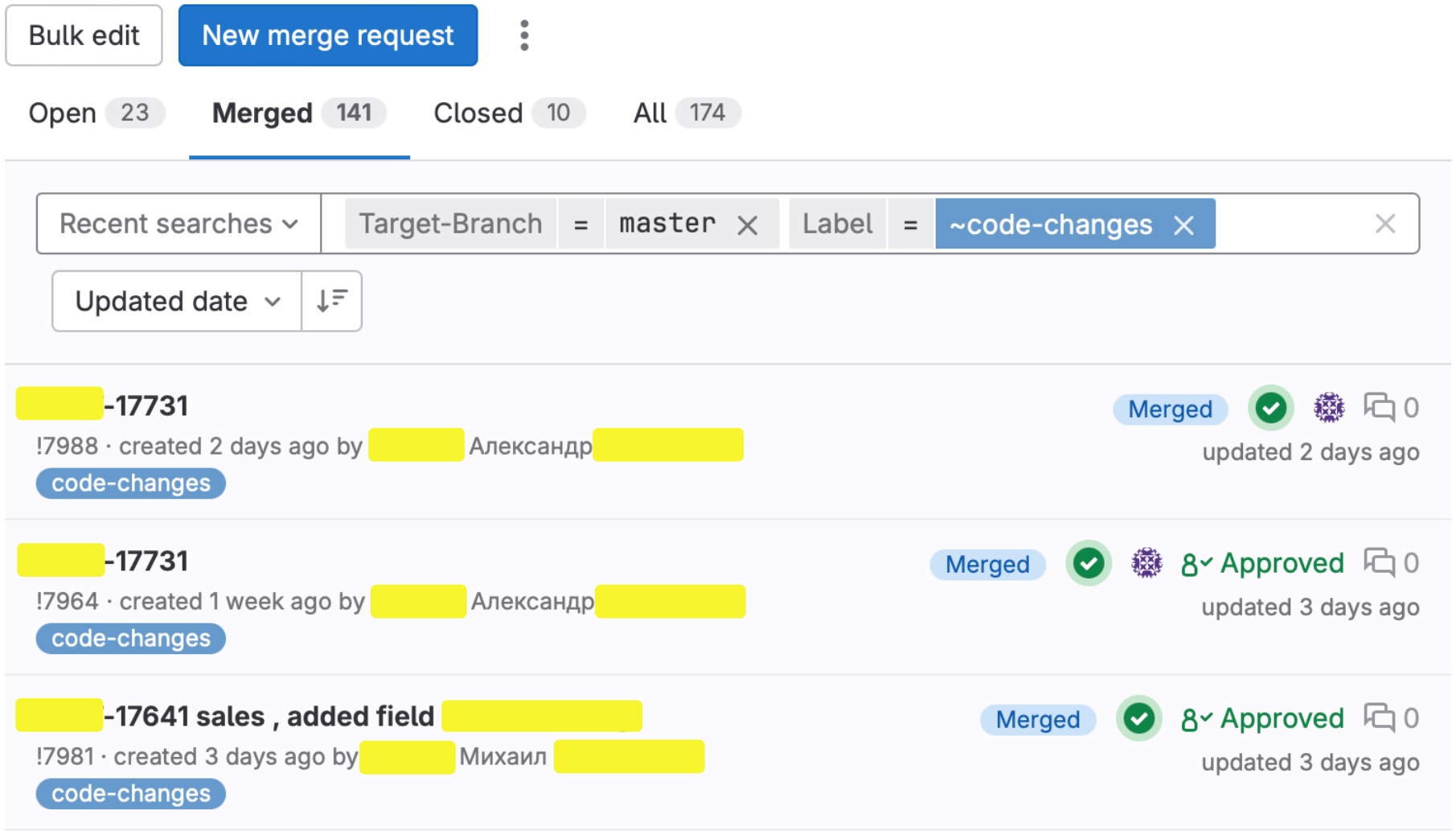
Процессы – release



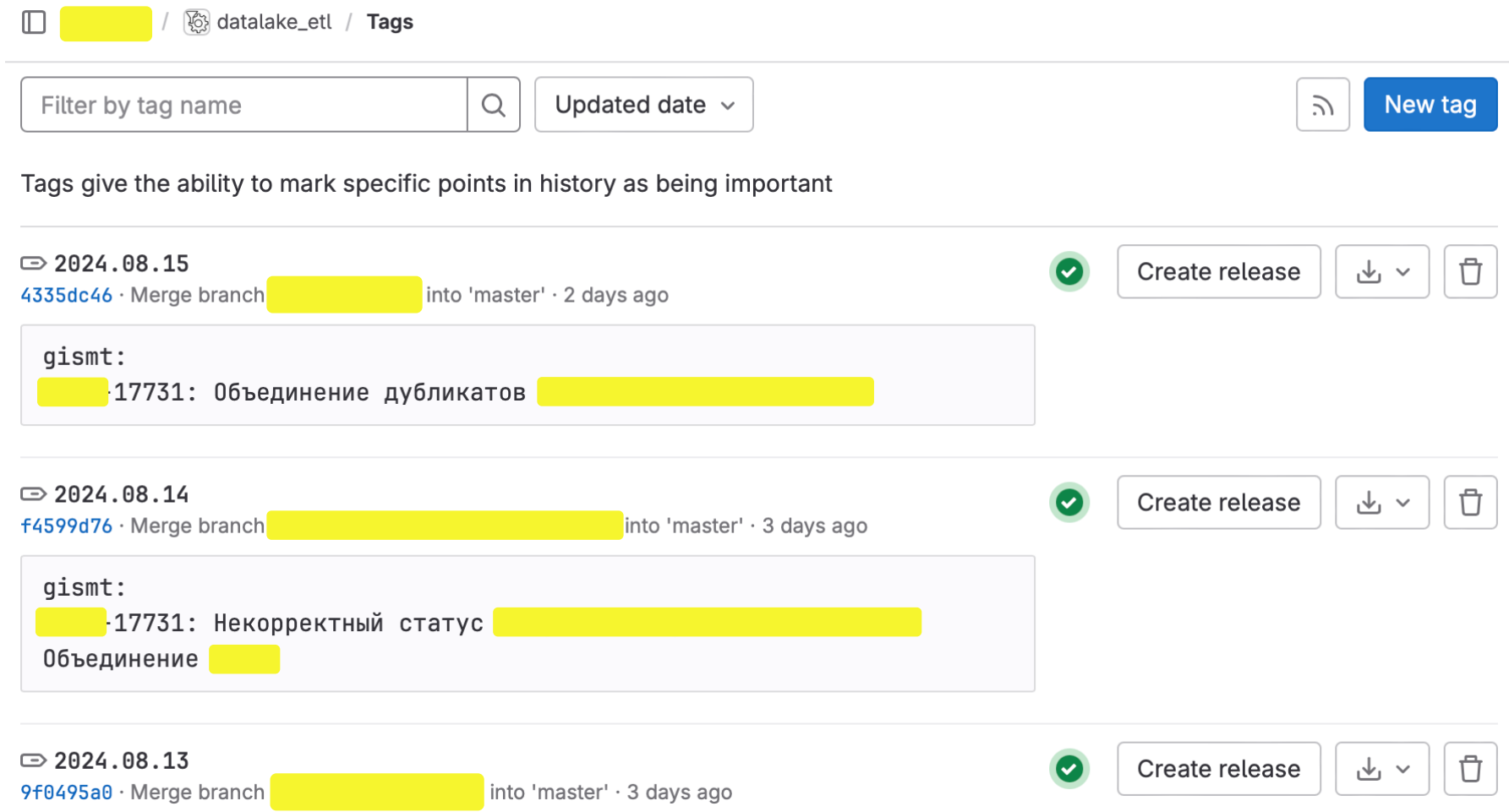
Процессы – release



Процессы – release



Процессы – release

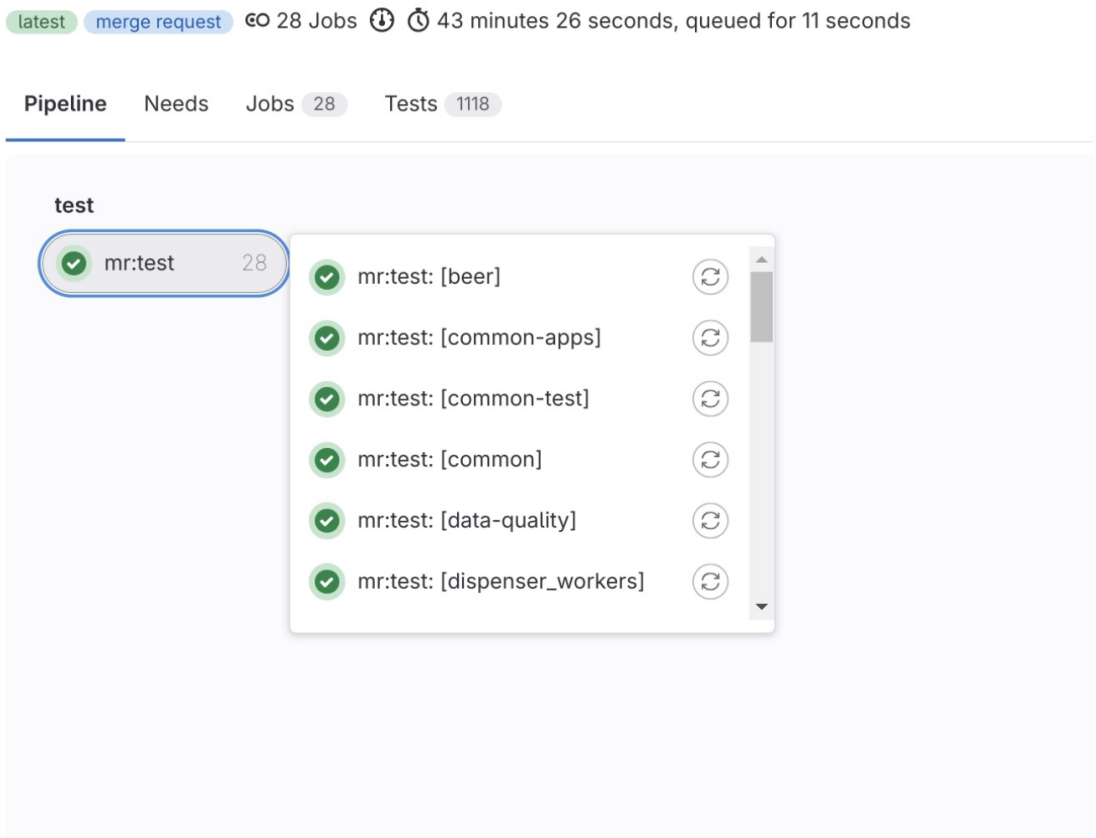


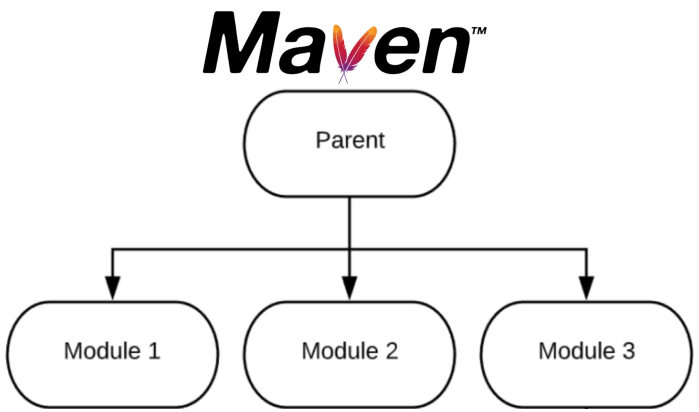
Pipelines

Pipelines

1. MR: test
2. Tag: test & build & deploy (prod)
3. Run pipeline manual: test & build & deploy (test)

Merge Request

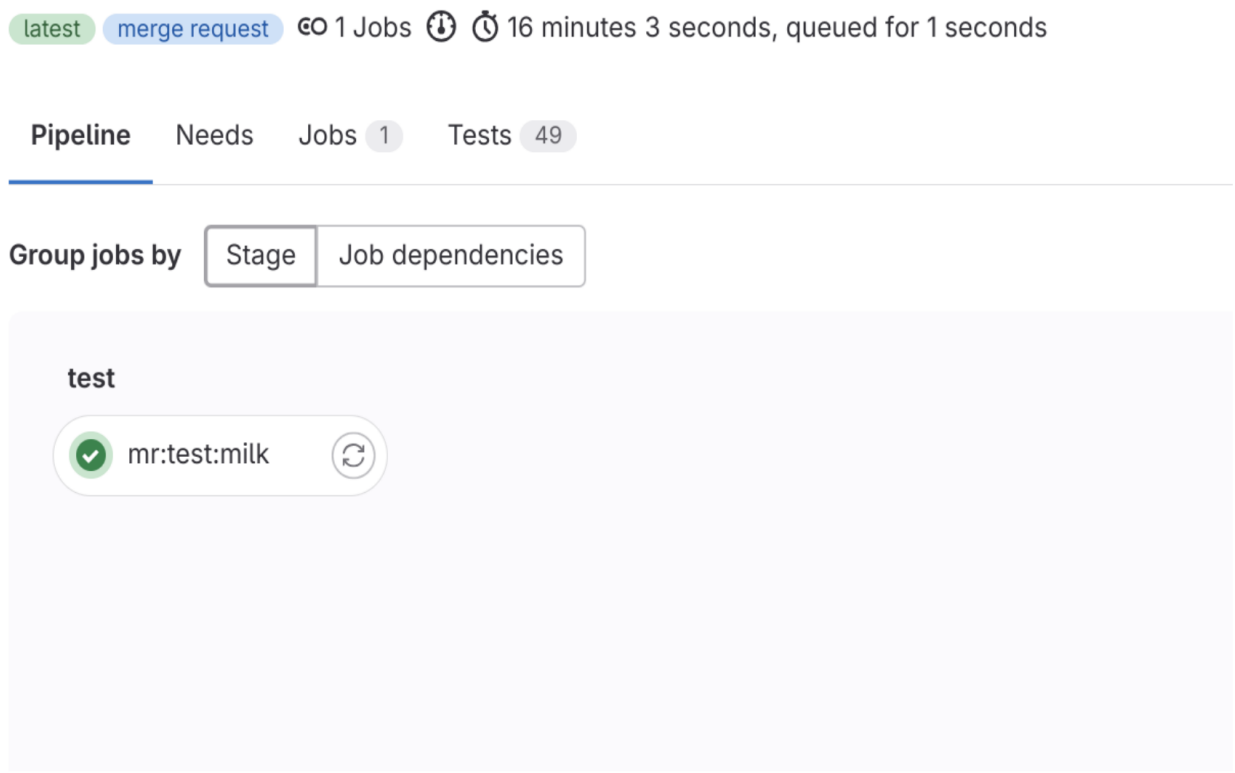




Example of `parallel:matrix`:

```yaml
deploystacks:
  stage: deploy
  script:
    - bin/deploy
  parallel:
    matrix:
      - PROVIDER: aws
        STACK:
          - monitoring
          - app1
          - app2
      - PROVIDER: ovh
        STACK: [monitoring, backup, app]
      - PROVIDER: [gcp, vultr]
        STACK: [data, processing]
  environment: $PROVIDER/$STACK
```

Merge Request

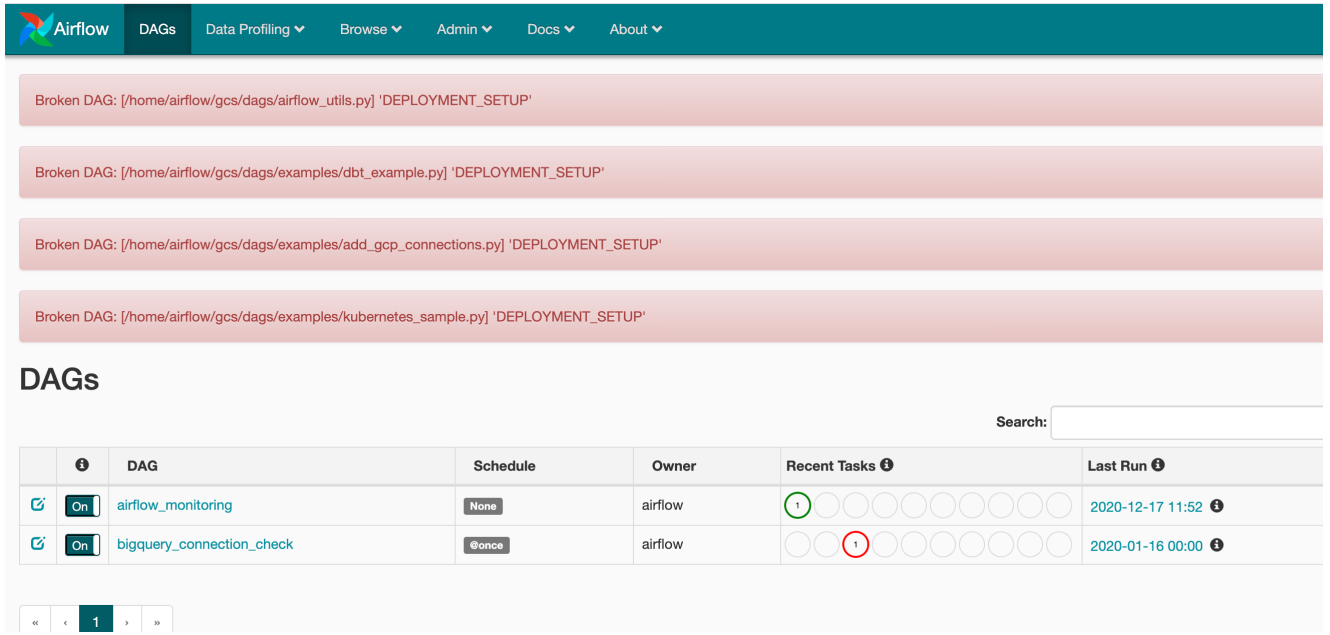


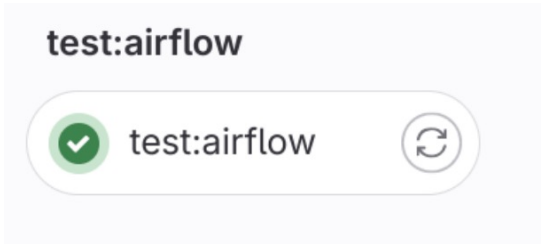
Example of `rules:changes`:

```yaml
docker build:
  script: docker build -t my-image:$CI_COMMIT_REF_SLUG .
  rules:
    - if: $CI_PIPELINE_SOURCE == "merge_request_event"
      changes:
        - Dockerfile
      when: manual
      allow_failure: true

docker build alternative:
  variables:
    DOCKERFILES_DIR: 'path/to/dockerfiles'
  script: docker build -t my-image:$CI_COMMIT_REF_SLUG .
  rules:
    - if: $CI_PIPELINE_SOURCE == "merge_request_event"
      changes:
        - $DOCKERFILES_DIR/**/*
```

Merge Request





variables

connections

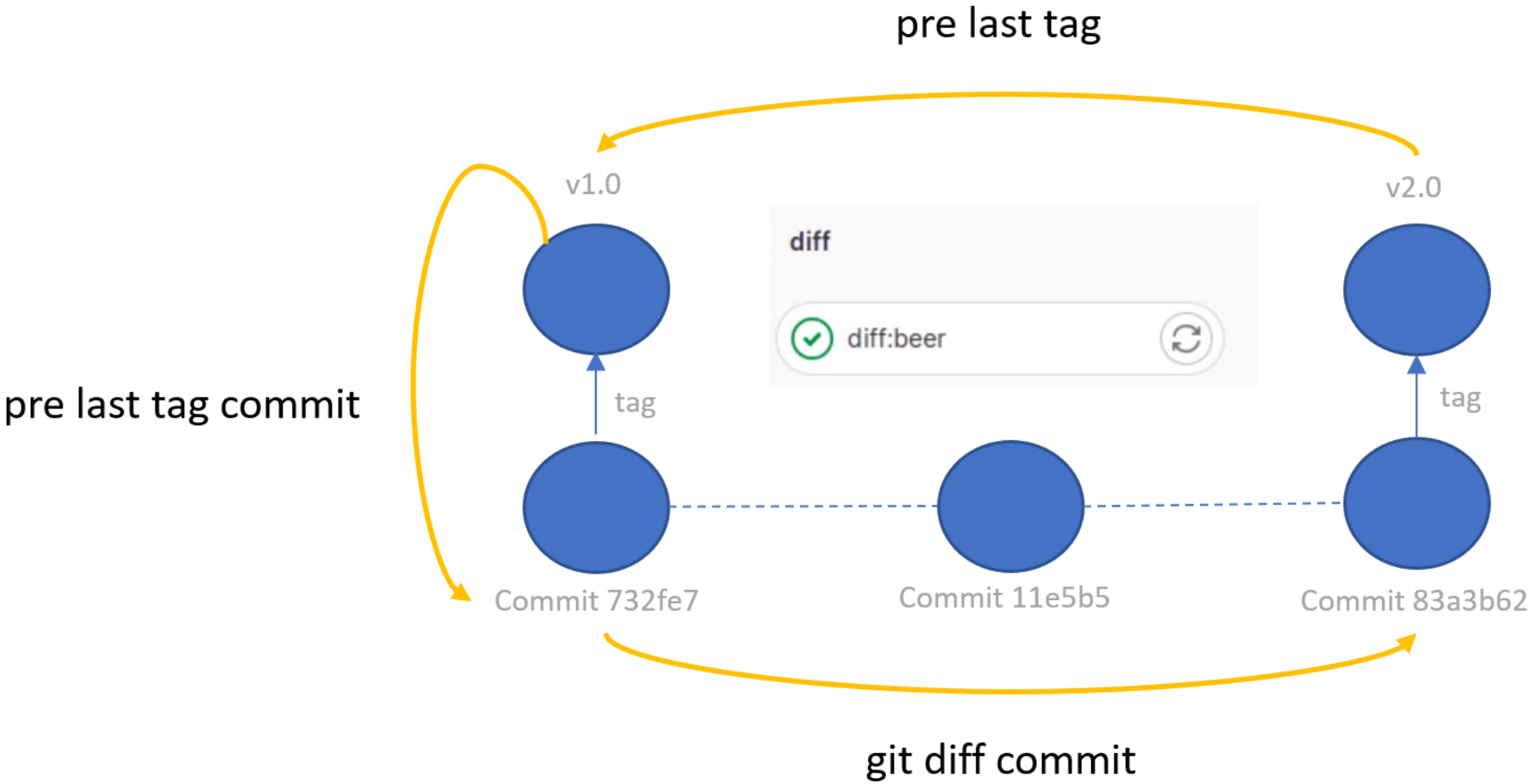
Airflow prod cluster

variables

connections

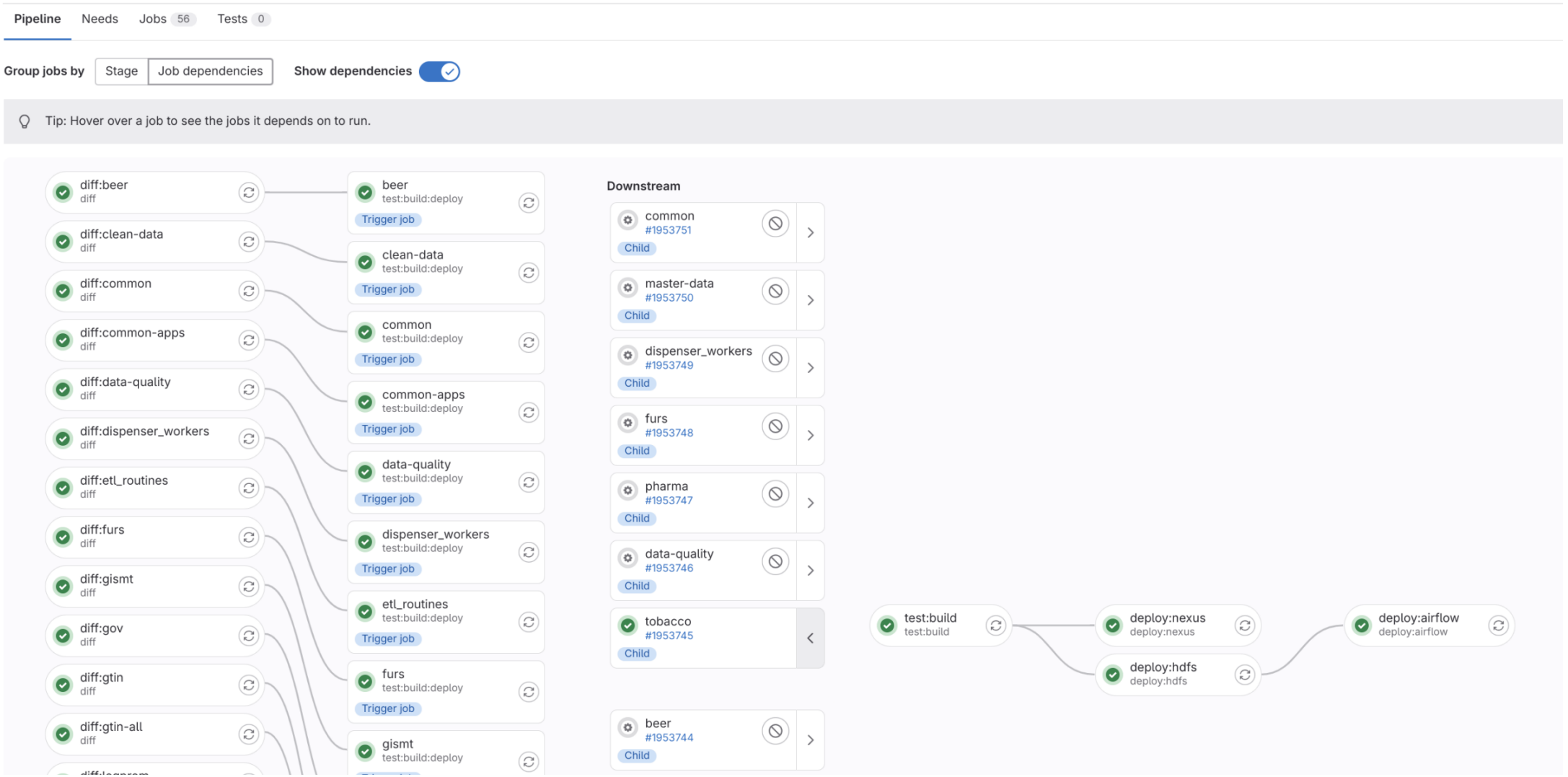
Smoke test Dag's

Tag

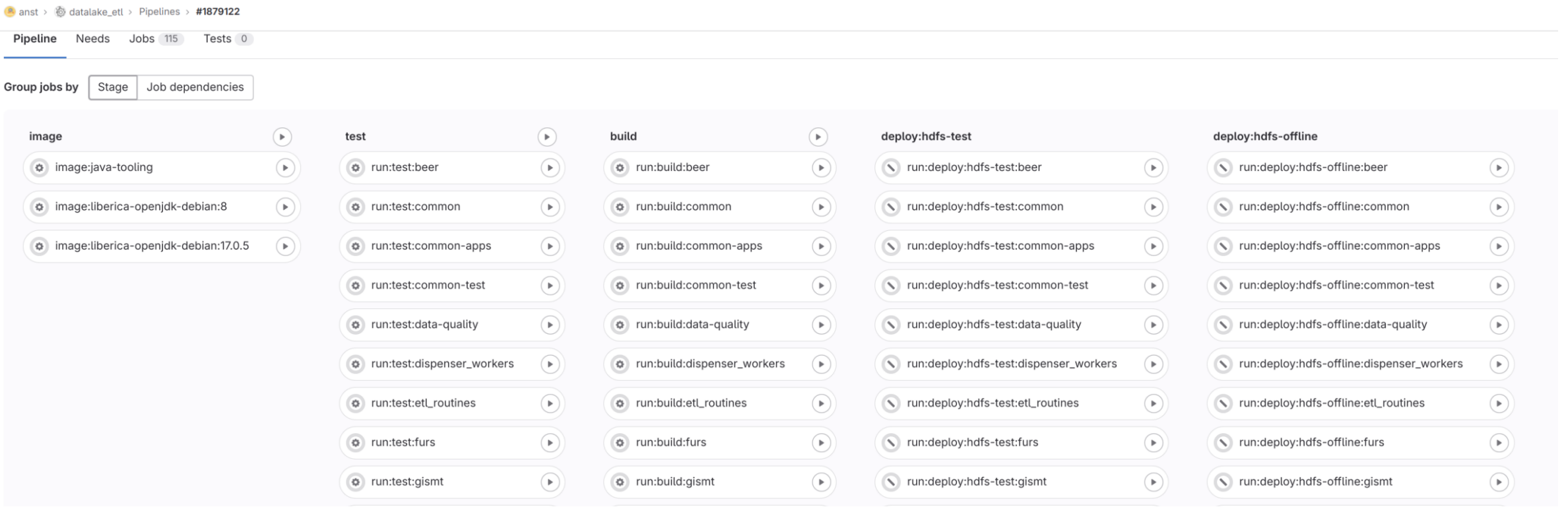


проверка на наличие изменений в модуле

Tag

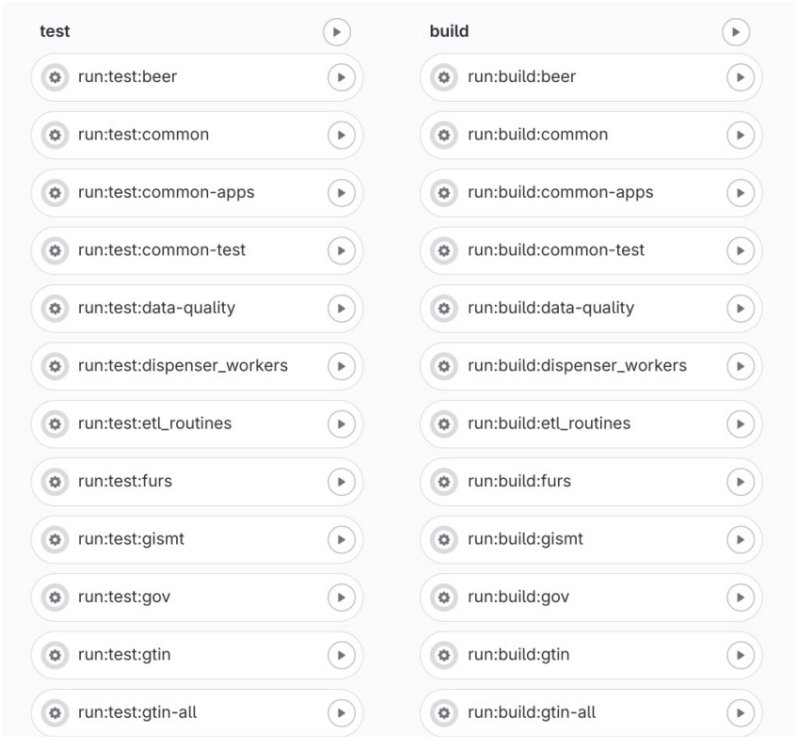

Pipeline
Needs
Jobs 56
Tests 0
Group jobs by
Stage
Job dependencies
Show dependencies
Tip: Hover over a job to see the jobs it depends on to run.
diff:beer
diff
diff:clean-data
diff
diff:common
diff
diff:common-apps
diff
diff:data-quality
diff
diff:dispenser_workers
diff
diff:etl_routines
diff
diff:furs
diff
diff:gismt
diff
diff:gov
diff
diff:gtin
diff
diff:gtin-all
diff
beer
test:build:deploy
Trigger job
clean-data
test:build:deploy
Trigger job
common
test:build:deploy
Trigger job
common-apps
test:build:deploy
Trigger job
data-quality
test:build:deploy
Trigger job
dispenser_workers
test:build:deploy
Trigger job
etl_routines
test:build:deploy
Trigger job
furs
test:build:deploy
Trigger job
gismt
test:build:deploy
Downstream
common
#1953751
Child
master-data
#1953750
Child
dispenser_workers
#1953749
Child
furs
#1953748
Child
pharma
#1953747
Child
data-quality
#1953746
Child
tobacco
#1953745
Child
beer
#1953744
Child
test:build
test:build
deploy:nexus
deploy:nexus
deploy:hdfs
deploy:hdfs
deploy:airflow
deploy:airflow

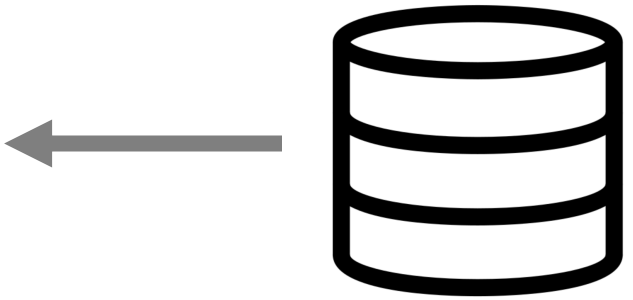
Manual


anst › datalake_etl › Pipelines › #1879122
Pipeline
Needs
Jobs 115
Tests 0
Group jobs by
Stage
Job dependencies
image
image:java-tooling
image:liberica-openjdk-debian:8
image:liberica-openjdk-debian:17.0.5
test
run:test:beer
run:test:common
run:test:common-apps
run:test:common-test
run:test:data-quality
run:test:dispenser_workers
run:test:etl_routines
run:test:furs
run:test:gismt
build
run:build:beer
run:build:common
run:build:common-apps
run:build:common-test
run:build:data-quality
run:build:dispenser_workers
run:build:etl_routines
run:build:furs
run:build:gismt
deploy:hdfs-test
run:deploy:hdfs-test:beer
run:deploy:hdfs-test:common
run:deploy:hdfs-test:common-apps
run:deploy:hdfs-test:common-test
run:deploy:hdfs-test:data-quality
run:deploy:hdfs-test:dispenser_workers
run:deploy:hdfs-test:etl_routines
run:deploy:hdfs-test:furs
run:deploy:hdfs-test:gismt
deploy:hdfs-offline
run:deploy:hdfs-offline:beer
run:deploy:hdfs-offline:common
run:deploy:hdfs-offline:common-apps
run:deploy:hdfs-offline:common-test
run:deploy:hdfs-offline:data-quality
run:deploy:hdfs-offline:dispenser_workers
run:deploy:hdfs-offline:etl_routines
run:deploy:hdfs-offline:furs
run:deploy:hdfs-offline:gismt

Cache

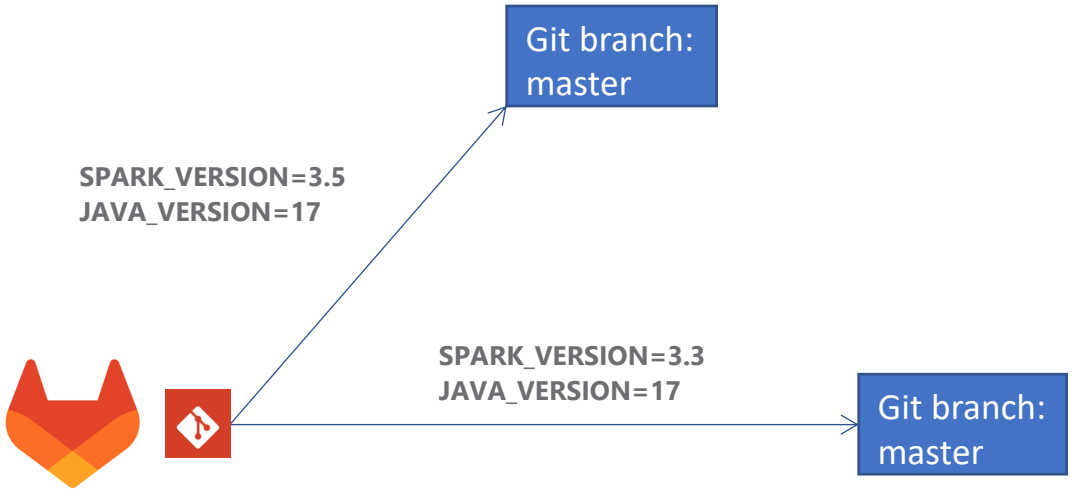
Maven cache & Gitlab CI/CD cache

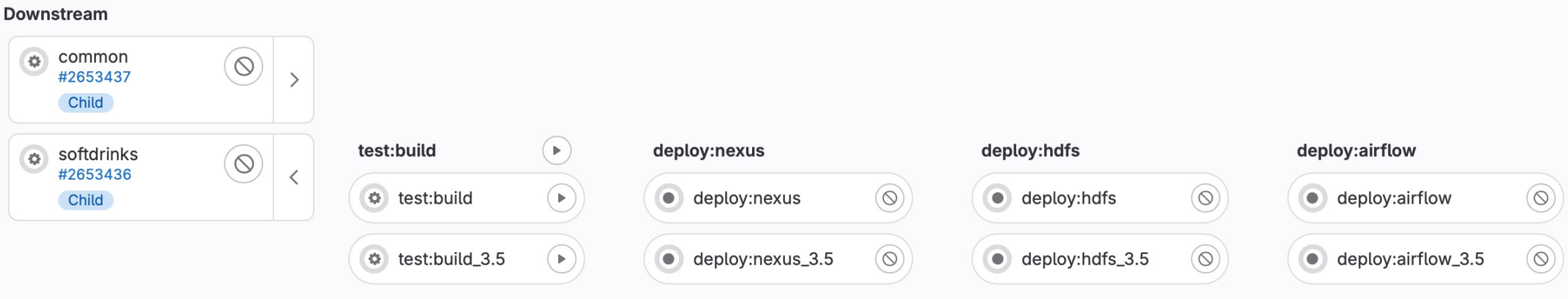




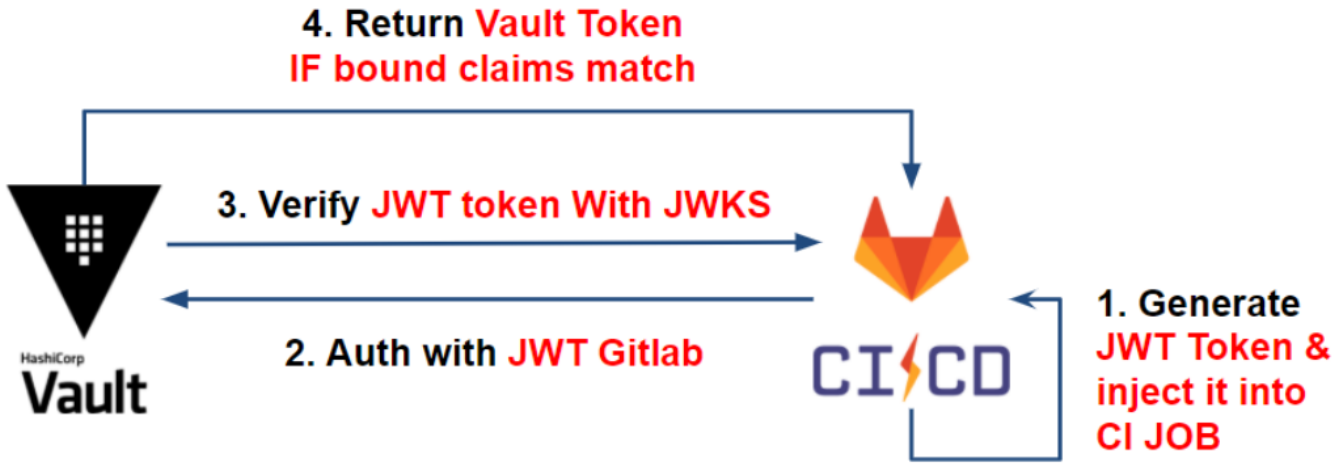
```
cache:
  key: master33cache
  policy: pull
  paths:
    - .m2/repository/
```

Spark


Git branch:
master
SPARK_VERSION=3.5
JAVA_VERSION=17
SPARK_VERSION=3.3
JAVA_VERSION=17
Git branch:
master


Downstream
common
#2653437
Child
softdrinks
#2653436
Child
test:build
test:build
test:build_3.5
deploy:nexus
deploy:nexus
deploy:nexus_3.5
deploy:hdfs
deploy:hdfs
deploy:hdfs_3.5
deploy:airflow
deploy:airflow
deploy:airflow_3.5

Security



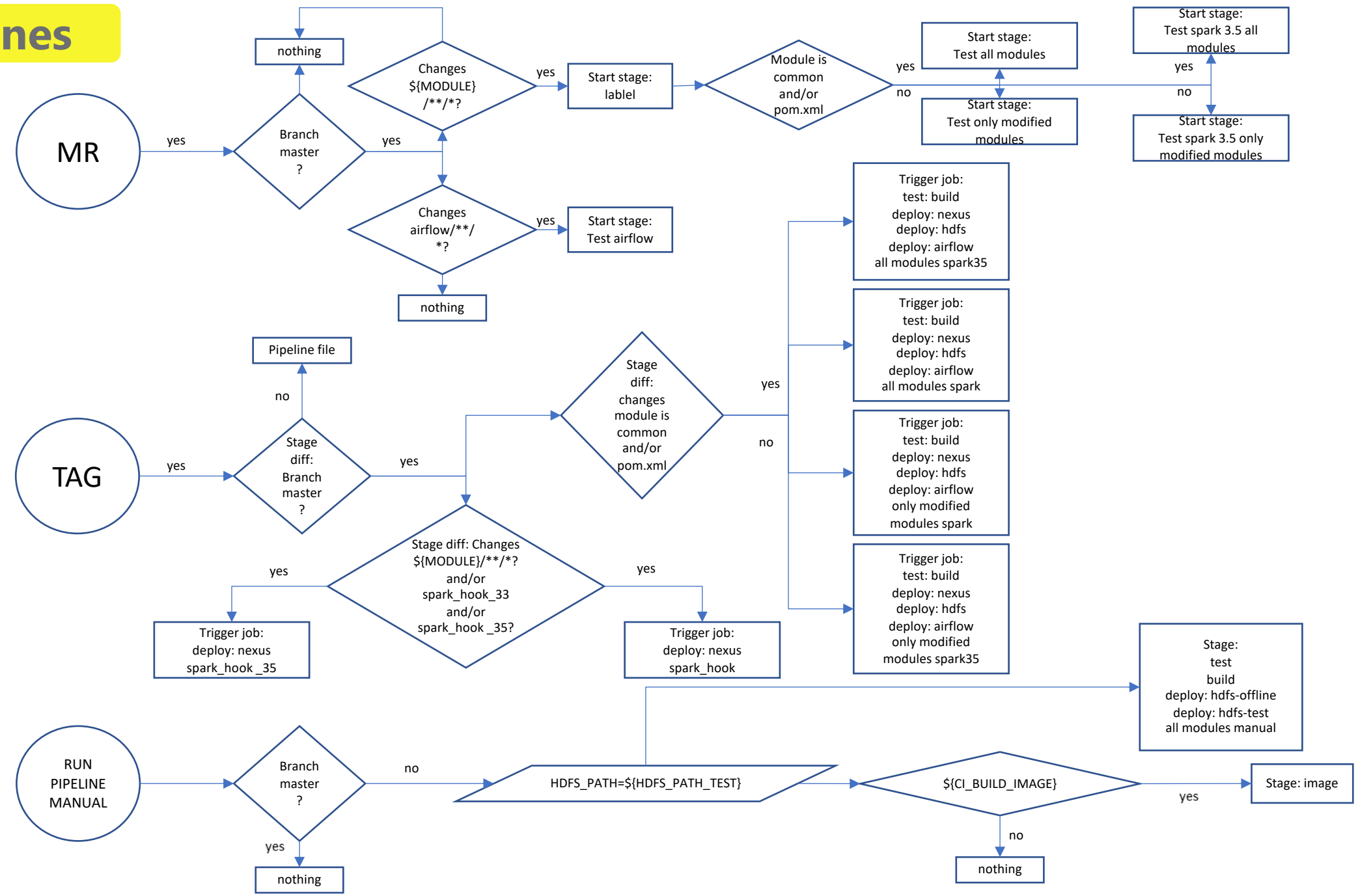
Members 3

Filter members

Account

Account	Source	Access granted	Max role	Expiration
Administrator It's you @root	demo	5 minutes ago	Owner	Expiration date
User 0 @user0	Inherited demo	1 minute ago by Administrator	Reporter	Expiration date
User 1 @user1	Direct member	54 seconds ago by Administrator	Guest	Expiration date

Pipelines


MR
yes
Branch master ?
yes
nothing
Changes ${MODULE} /**/*?
yes
Start stage: lablel
Module is common and/or pom.xml
yes
Start stage: Test all modules
no
Start stage: Test only modified modules
Start stage: Test spark 3.5 all modules
yes
no
Start stage: Test spark 3.5 only modified modules
Changes airflow/**/ *?
yes
Start stage: Test airflow
nothing
TAG
yes
Stage diff: Branch master ?
no
Pipeline file
yes
Stage diff: changes module is common and/or pom.xml
yes
no
Trigger job: test: build deploy: nexus deploy: hdfs deploy: airflow all modules spark35
Trigger job: test: build deploy: nexus deploy: hdfs deploy: airflow all modules spark
Trigger job: test: build deploy: nexus deploy: hdfs deploy: airflow only modified modules spark
Trigger job: test: build deploy: nexus deploy: hdfs deploy: airflow only modified modules spark35
Stage diff: Changes ${MODULE}/**/*? and/or spark_hook_33 and/or spark_hook _35?
yes
yes
Trigger job: deploy: nexus spark_hook _35
Trigger job: deploy: nexus spark_hook
RUN PIPELINE MANUAL
Branch master ?
no
yes
nothing
HDFS_PATH=${HDFS_PATH_TEST}
Stage: test build deploy: hdfs-offline deploy: hdfs-test all modules manual
${CI_BUILD_IMAGE}
yes
Stage: image
no
nothing

Оптимизации I/O

Оптимизации I/O

```
val res = result.columns.foldLeft(
  .select(columns.map(col): _*)

res.toJSON.show(false)    You, 25.0

val expDf = spark.read.json(getCla
val expected = expDf.columns.foldL
  .select(columns.map(col): _*)

checkAnswer(res, expected)
```

Оптимизации I/O

```
worker:a doing lots of uninteresting work +0ms
worker:b doing some work +0ms
http listening +23ms
worker:a doing lots of uninteresting work +424ms
worker:a doing lots of uninteresting work +307ms
worker:b doing some work +814ms
worker:b doing some work +58ms
worker:a doing lots of uninteresting work +312ms
worker:a doing lots of uninteresting work +647ms
worker:a doing lots of uninteresting work +469ms
worker:b doing some work +1s
worker:a doing lots of uninteresting work +797ms
worker:a doing lots of uninteresting work +153ms
worker:b doing some work +1s
worker:a doing lots of uninteresting work +491ms
worker:a doing lots of uninteresting work +323ms
worker:b doing some work +602ms
```

Что дальше?

Развитие CI/CD

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    <build>
        <plugins>
            <!-- enable scalatest -->
            <plugin>
                <groupId>org.scalatest</groupId>
                <artifactId>scalatest-maven-plugin</artifactId>
                <version>${scalatest.maven.plugin.version}</version>
                <configuration>
                    <reportsDirectory>${project.build.directory}/surefire-reports</reportsDirectory>
                    <junitxml>.</junitxml>
                    <filereports>TestSuite.txt</filereports>
                    <!--Parallel doesn't work - Only one SparkContext should be running in this JVM (see SPARK-2243)-
                    <parallel>false</parallel>
```

Post scriptum

Post Scriptum

Мы ищем Data Engineer-ов на наш стек. Приходите ко мне в Telegram @nblagodarnyy

Post Scriptum

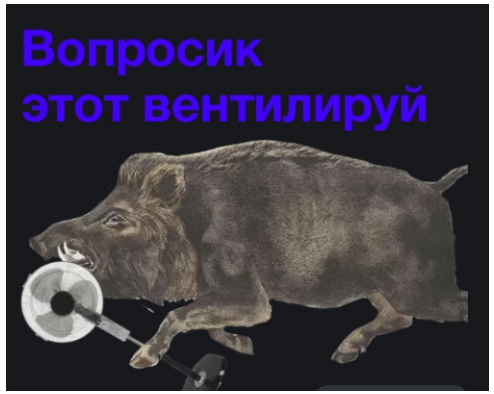
DE-meetup в СПБ 11/09

Ссылки

- https://www.baeldung.com/maven-dependencymanagement-vs-dependencies-tags
- https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html#Dependency_Management
- https://maven.apache.org/maven-ci-friendly.html

Post Scriptum

Вопросы?



СПАСИБО ЗА ВНИМАНИЕ!

Никита Благодарный
telegram @nblagodarnyy

Александра Чекмарева (Китченко)
telegram @sasha_kitchen

ЧЕСТНЫЙ ЗНАК | центр развития перспективных технологий

SmartData 2024

Post Scriptum

DE-meetup в СПБ 11/09