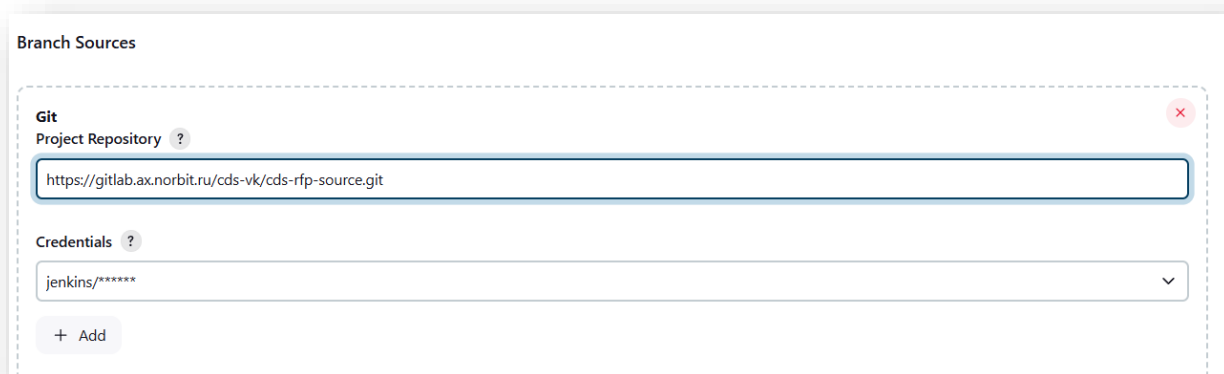


ИНСТРУКЦИЯ ПО УСТАНОВКЕ

В качестве CI-CD инструмента используется Jenkins, Jenkinsfile для каждого сервиса расположен в дополнительной папке - cicd

Для каждого сервиса создаётся Job – Multibranch Pipeline, в настройках которой нужно только ввести ссылку на Git репозиторий и выставить параметр Property strategy – Named branches get different properties



В Jenkins в глобальных переменных должны быть указаны параметры адрес, логин и пароль от MAVEN репозитория:

- docker_registry_address
- *****ADDRESS***:***PORT*****

- maven_login
- *****LOGIN*****

- maven_pass
- *****PASSWORD*****

Версии используемых базовых Docker образов прописаны в Dockerfile папке cicd для каждого проекта.

Для подключения к nexus на серверах кластера k8s необходимо прописать адрес сервера nexus в файл /etc/containerd/config.toml

Версии используемых базовых Docker образов прописаны в values.yaml helm-чартах каждого проекта.

Для всего проекта нужен postgresql сервер, если требуется его поднять внутри кластера kubernetes то можно использовать готовый helm шаблон, стоит только сменить пароли указанные на строках 28, 142, 151, 152, 153 и если требуется сменить nodePorts то его можно поменять на строке 592 в файле values.yaml в папке с helm шаблоном в каталоге cicd. Для разворачивания postgresql сервера нужно скопировать папку с helm шаблоном на ноду кластера и из неё выполнить команду:

- `helm install -f values.yaml -n ***NAMESPACE***`

Сервер развернётся в нужном namespace kubernetes кластера.

Так же можно использовать отказоустойчивые решения типа patroni, позволяющие организовать отказоустойчивый кластер postgresql

auth-server

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-auth-source>

Сборка образа и его последующий деплой в kubernetes. В папке cicd/auth-server/ есть пример pom.xml, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

cds-dr-updater

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-dr-update-source>

Сборка образа. В папке cicd/cds-dr-updater/ есть пример pom.xml, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

cds-fulltext-search

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-fulltext-search-source>

Сборка образа и его последующий деплой в kubernetes. В папке cicd/cds-fulltext-search/ есть пример pom.xml, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

cds-memcached

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-memcached>

Делой memcached в kubernetes для проекта.

frontend

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-frontend-source>

Деплой в kubernetes сервиса cds-ui используя образ созданный после работы Job - cds-web-api.

planning-time

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-planning-time-source>

Сборка образа planning-time. В папке cicd/planning-time/ есть пример pom.xml, в нём нужно прописать адрес nexus сервера на который будет заливаться готовый образ после сборки.

cds-android

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-phone-application-source>

Сборка APK приложения для android. Для сборки нужен образ в котором

В файле Jenkinsfile в stage «Build Android Apk» требуется образ содержащий android sdk, gradle, java в каталоге cicd/cds-android/ лежит Dockerfile-ubuntu с инструкциями для сборки этого образа. Готовый образ следует залить в nexus для использования его при сборке зависимостей. В строке 79 в Jenkinsfile как раз указан этот кастомный образ. Далее в stage «Build nginx image with Apk» собирается образ nginx в который кладётся APK, после в stage «Deploy image in Kubernetes» этот образ деплоится в kubernetes как web страничка с которой можно скачать по прямой ссылке этот готовый APK.

cds-rfp

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-rfp-source>

Сборка образа и его последующий деплой в kubernetes. В папке `сiсd/cds-rfp/` есть пример `rom.xml`, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

cds-persistence

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-planning-time-source>

После установки postgresql сервера для проливки миграций БД. Это первая job которую нужно запустить перед всеми остальными.

cds-notify

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-notify-source>

Сборка образа и его последующий деплой в kubernetes. В папке `сiсd/cds-notify/` есть пример `rom.xml`, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

cds-print

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-print-source>

Сборка образа и его последующий деплой в kubernetes. В папке `сiсd/cds-print/` есть пример `rom.xml`, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

cds-proto

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-proto-source>

Сборка ключевых зависимостей сервисов. Эту Job стоит запускать до сборки и деплоя остальных сервисов. В папке `сiсd/cds-proto/` есть пример `rom.xml`, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

cds-route-optimizer

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-route-optimizer-source>

Сборка образа и его последующий деплой в kubernetes. В папке `сiсd/cds-route-optimizer/` есть пример `rom.xml`, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

В kubernetes для того чтобы Jobs могли выкачивать из nexus репозитория готовые образы для деплоя нужно создать `secret`, где прописан адрес этого репозитория(в BASE64 кодировке). Пример манифеста лежит в `сiсd/deploy/` файл `regcred.yaml`. Установка этих манифестов выглядит так:

- `kubectl apply -f regcred.yaml`

cds-hive

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-saas-source>

Сборка образа и его последующий деплой в kubernetes. В папке `сiсd/cds-route-optimizer/` есть пример `rom.xml`, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

В kubernetes для того чтобы Jobs могли выкачивать из nexus репозитория готовые образы для деплоя нужно создать `secret`, где прописан адрес этого репозитория(в BASE64 кодировке). Пример манифеста лежит в `сiсd/deploy/` файл `regcred.yaml`. Установка этих манифестов выглядит так:

- `kubectl apply -f regcred.yaml`

cds-infra

GIT: <https://gitlab.marshrur.ru/cds-vk/cds-infra>

Содержит Dockerfile всех сервисов, настройки grpc-envoy, osrm и БД.

Сборка образа и его последующий деплой в kubernetes. В папке `сiсd/cds-route-optimizer/` есть пример `rom.xml`, в нём нужно прописать адрес nexus сервера из которого будут подтягиваться зависимости для сборки проекта.

В kubernetes для того чтобы Jobs могли выкачивать из nexus репозитория готовые образы для деплоя нужно создать `secret`, где прописан адрес этого репозитория(в BASE64 кодировке). Пример манифеста лежит в `сiсd/deploy/` файл `regcred.yaml`. Установка этих манифестов выглядит так:

- `kubectl apply -f regcred.yaml`