Сборка и внутренний деплой нового образа Tarantool

Общая информация

Tarantool - система производительных компьютерных вычислений и распределенного кэширования. См. GitHub https://github.com/tarantool/tarantool.

Tarantool совмещает в себе базу данных и сервер приложений: логика обработки данных описывается на языке lua, который преобразуется в машинный код при исполнении.

Для целей быстроты работает в памяти (in-memory), но также имеет и дисковый движок для данных, перенесенных из RAM на диск.

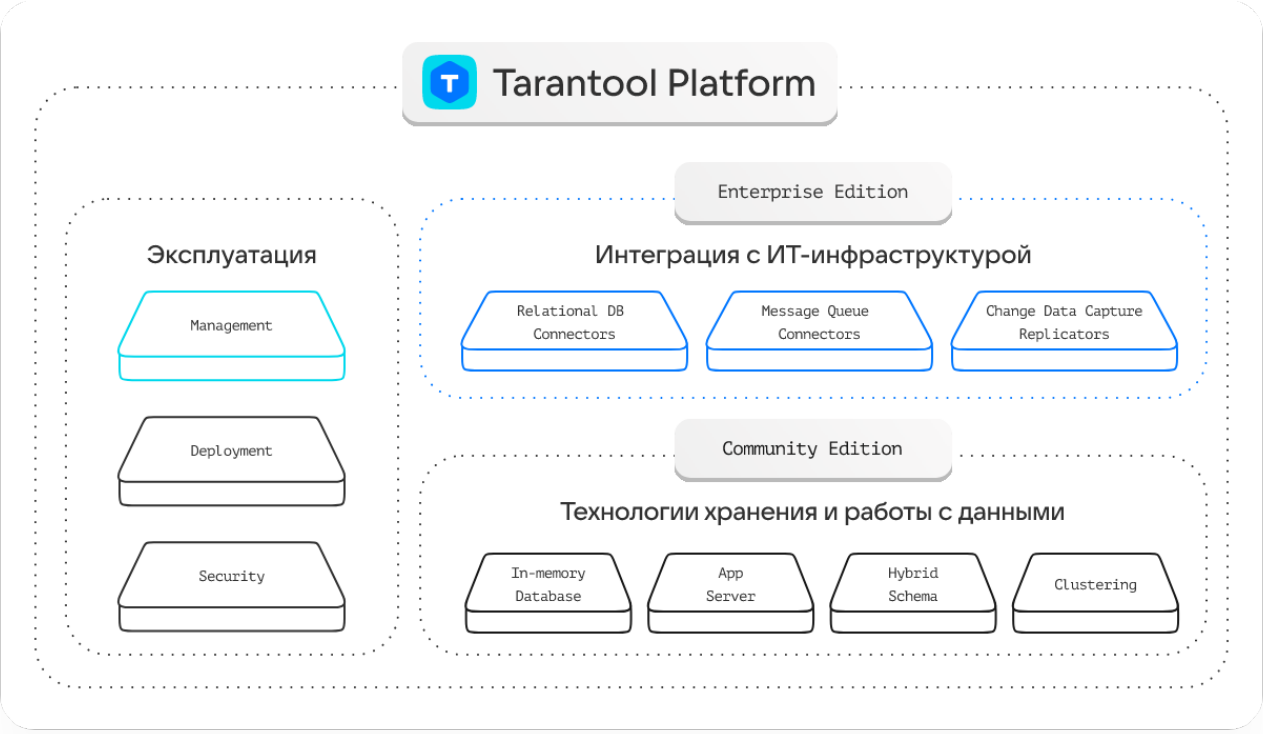
Для создания приложения в распределенном виде, используется специальный фреймворк для масштабирования под названием cartridge. См. GitHub https://github.com/tarantool/cartridge.С помощью него можно сделать несколько копий на разных серверах или разделить данные по нескольким узлам Tarantool, создавать и настраивать кластеры из нескольких экземпляров Tarantool.

В целом, **tarantool не является аналогом memcached или redis, а обладает более богатым функционалом** (см. документацию и GitHub по ссылкам выше).

На проекте VTBMSA tarantool используется в качестве "персистентного кэша" из-за своих преимуществ:

- Быстрота: Чувствительный к быстродействию функционал VTBMSA полагается на содержимое кэша Tarantool;
- Надежность: Есть гарантии сохранности, при критических ситуациях содержимое кэша должно оставаться сохранным.

Используется Enterprise версия, так как она включает нужный функционал:



Где скачать и источники информации

Скачать Tarantool SDK можно с сайта https://www.tarantool.io. Для этого нужна регистрация. Можно регистрироваться с помощью электронной почты Синимекс с указанием Банка ВТБ (АО) в качестве компании.

Есть рабочие чаты: "(Cinimex+VK_без ВТБ)Tarantool_рабочие вопросы" и официальный канал Tarantool https://t.me/tarantoolru.

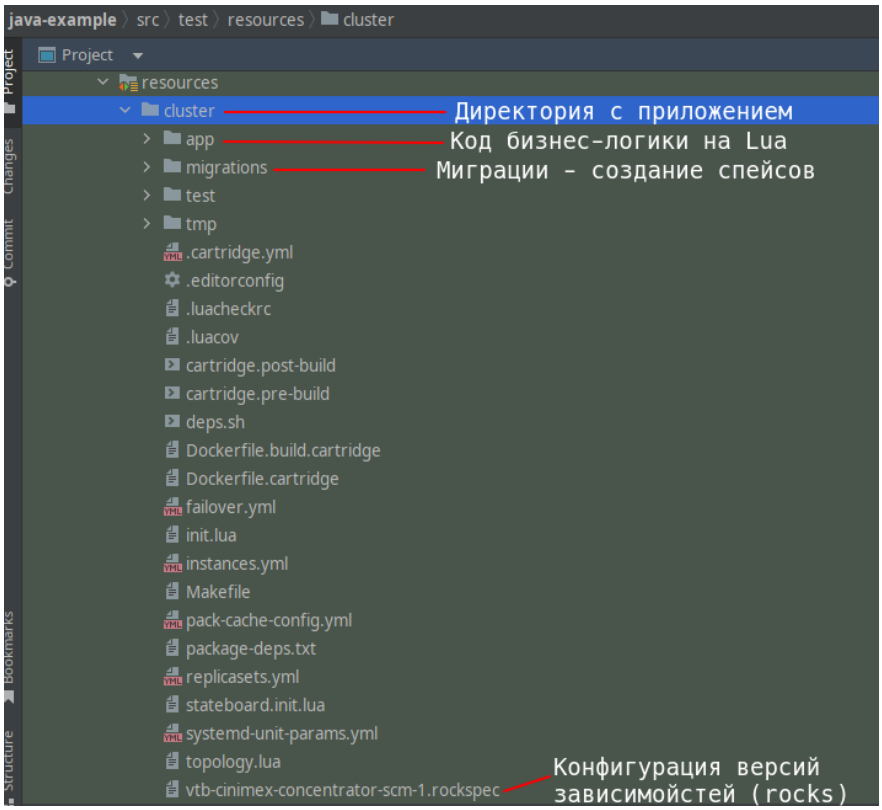
Компетенции по Tarantool закреплены за Отделом интегрированных систем, руководитель **Кадриев Алмаз**.

Обученные спецы по Tarantool **Паленный Дмитрий** и **Луценко Володя.**

Наше приложение (правка кластера Tarantool)

Для использования Tarantool нужно приложение, к сожалению, написанное на языке lua.

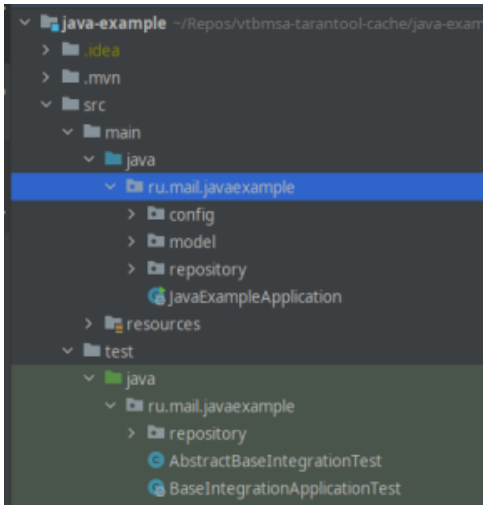
Чтобы изменить спейсы Tarantool, нужно внести изменения в репозитории https://src.cinimex.ru/VTBMSA/vtbmsa-tarantool-cache
Миграции, хранимые процедуры и т.п. содержатся в Maven-проекте java-example. Примерная структура кластера показана на рисунке ниже:



Для внесения правок в схеме данных Tarantool, нужно править именно этот код. Например, если добавляем новое поле в какой-либо спейс, нужно поправить файл миграции этого спейса.

После этого нужно также внести изменения в java-код проекта java-example: поправить entity-сущности в пакете model, репозитории, а также юнит-тесты.

Интеграционный тест ru.mail.javaexample.AbstractBaseIntegrationTest можно прогонять не на контейнере Тарантул, а на локальном докер-контейнере. Для этого закомментировать создание контейнера в этом классе.



Внести запись в changelog репозитория https://src.cinimex.ru/VTBMSA/vtbmsa-tarantool-cache с описанием правок и версией образа.

На этом правка Tarantool кластера завершена.

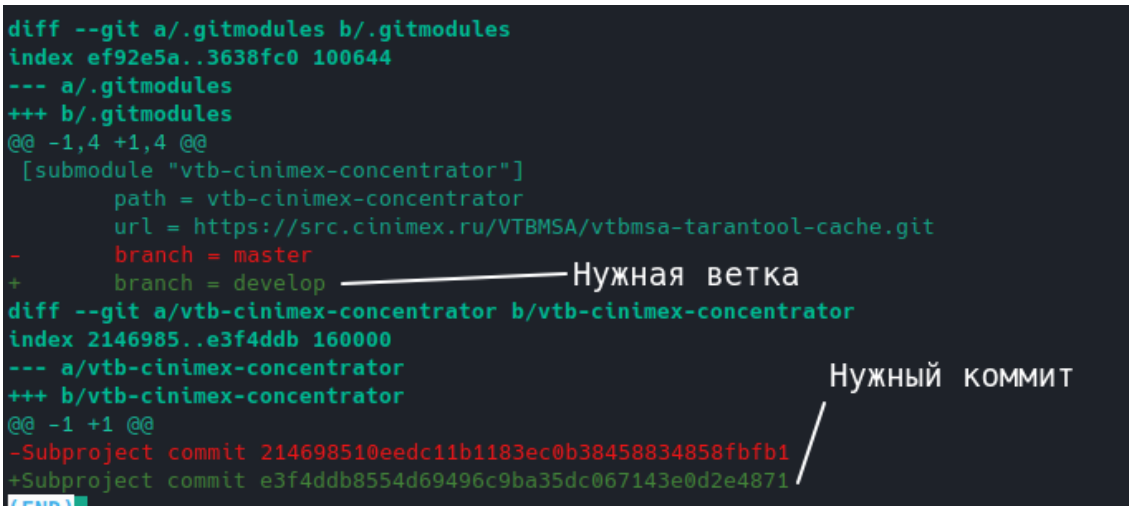
Сборка докер-образа измененного кластера Tarantool

Для сборки докер-образа измененного кластера Tarantool используется Dockerfile в **модульном репозиторий** Git https://src.cinimex.ru/VTBMSA/tarantool-container.
В нем содержится ССЫЛКА на репозиторий https://src.cinimex.ru/VTBMSA/vtbmsa-tarantool-cache в виде модуля "vtb-cinimex-concentrator".

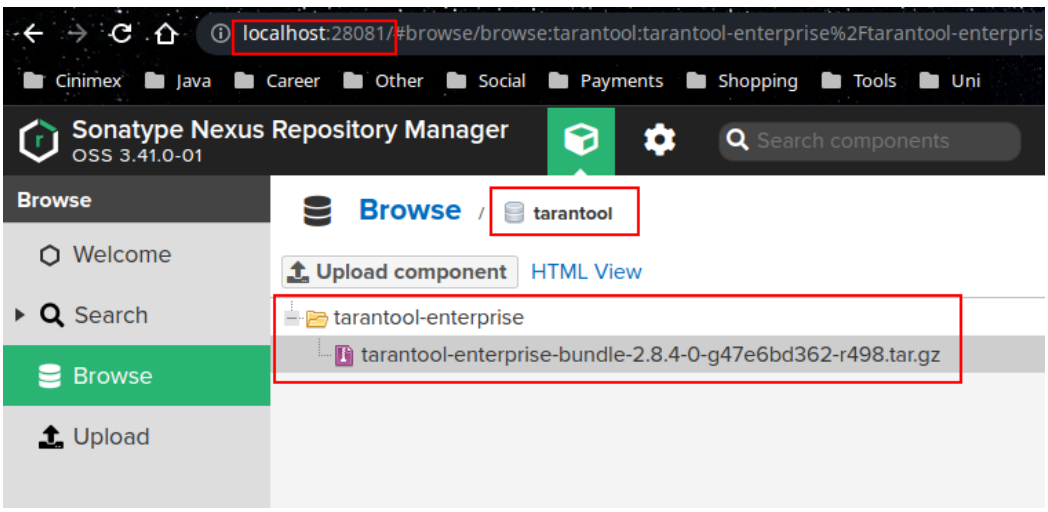
Перед сборкой образа нужно проверить, подгружены ли изменения в модуле (какой коммит сейчас видит модуль по ссылке), а также проверить, правильная ли ветка указана для этого модуля.

Как переключить git submodule на другую ветку:
1 Правим файл .gitmodules вручную или командой "git config -f .gitmodules".
2 Синхронизировать эти изменения в .git/config командой:
git submodule sync
3 Обновить модули командой:
git submodule update --init --remote

Видим, что указана нужная ветка и нужный коммит, чтобы сборка образа выполнялась из нужного кластера Tarantool:



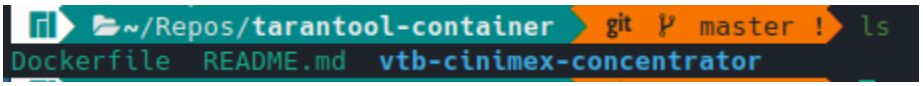
1. Развернуть сервер sonatype/nexus3 локально (можно и не локально, неважно). Хорошая статья: https://voltwu.github.io/blog/docker/2021/01/21/using-nexus3-as-your-repository-a-simple-guide/
В nexus3 в raw-репозитории загрузить SDK нужной версии, например: tarantool-enterprise-bundle-2.8.4-0-g47e6bd362-r498.tar.gz



Принципиально, способ хостинга SDK Tarantool Enterprise значения не имеет, можно разместить его в любом месте, откуда скрипт сборки сможет выкачать его через wget.

2. Собрать новый образ tarantool с указанием версии образа <версия>-<минорная версия>-<hash коммита из семи цифр>, например: 0.2.18-0-e3f4ddb.
docker build --no-cache --build-arg TARANTOOL_URL_SDK=http://localhost:28081/repository/tarantool/tarantool-enterprise/tarantool-enterprise-bundle-2.8.4-0-g47e6bd362-r498.tar.gz --tag artifactory.cinimex.ru/vtbmsa_docker/vtbmsa/vtb-cinimex-concentrator:0.2.18-0-e3f4ddb .

Эта команда выполнит сборку докер-образа кластера Tarantool, описанную в Dockerfile, ее нужно выполнять в директории с Dockerfile:



3. Запустить собранный новый образ в контейнере и ПРОВЕРИТЬ работу тестами в проекте java-example, а также, можно позже, в приложении vtbmsa-lib-cache-rest (https://src.cinimex.ru/VTBMSA/vtbmsa-lib-cache-rest). Команда для запуска образа:
docker run -p 8081:8081 -p 3301:3301 -t -i -d --name tarantool artifactory.cinimex.ru/vtbmsa_docker/vtbmsa/vtb-cinimex-concentrator:0.2.18-0-e3f4ddb

Деплой на внутреннем тестовом сервере Cinimex

Новый образ tarantool нужно развернуть на тестовом стенде Ciminex, который используется во внутреннем тестировании (сейчас это сервер vtbmsa-05.vm.cmx.ru, но может меняться).

1. Предупредить всех заинтересованных лиц (тестировщиков), что tarantool будет недоступен и все данные будут удалены. Также будут удалены все данные в БД Postgresql.
После обработки возражений, остановить все сервисы, которые завязаны на tarantool в правильном порядке. Это делается установкой количества активных подов в значение "0" для соответствующего сервиса на сервере OpenShift, после чего поды будут погашены и не будут автоматически стартовать.
Очистить схемы в БД в правильном порядке с помощью drop cascade (doc → lau → dict).

2. С помощью удобного SSH-клиента (например, для ОС Windows это может быть MobaXTerm) подключаемся по протоколу ssh к нужному тестовому серверу tarantool.
Нужно подключаться через SSH
Креды:
host: vtbmsa-05.vm.cmx.ru
user: root

Пароль секретный, узнать у кого-нибудь из разработчиков или DevOps.

3. На сервере обновляем переменные среды для ssh-сессии:
source /sdk/tarantool-enterprise/env.sh
Проверяем, что среда оболочки изменена:
cartridge --version
*должен быть вывод с информацией по версии

4. Выгрузить обновления из git-репозитория
cd /apps/git/vtbmsa-tarantool-cache
git pull
!-Проверить, что все ОК синхронизировано с Git.

5. Собрать и запустить на сервере в режиме демона новый образ кластера Tarantool
cd /apps/cluster

*после каждой команды можно проверять статус узлов кластера командой **cartridge status**.

остановить узлы кластера Tarantool: **cartridge stop**
отчистить ссылки скриптов: **cartridge clean**

*если нужно использовать новую версию SDK Tarantool для сборки образа, то залить ее в директорию /sdk и распаковать командой **tar -xzf *.tar.gz.** Новая версия SDK Tarantool нужна, когда мы меняем версии зависимостей в файле *.rockspec нашего кластера, и этих rock-плагинов нет в старой версии SDK в /sdk/tarantool-enterprise/rocks.

собираем кластер: **cartridge build**
запускаем кластер в режиме демона: **cartridge start -d**

6. Сконфигурировать реплики и накатить спейсы:
создать реплики и шарды: **cartridge replicasets setup --bootstrap-vshard**
накатить миграции: **curl -X POST http://localhost:8081/migrations/up**

7. Запустить снова поды сервисов и проследить, что миграции накатились и все работает.

8. Разместить новый образ в артифактори Синимекс и обновить информацию о версии образов в Вики https://wiki.cinimex.ru/pages/viewpage.action?pageId=100378706
docker login https://artifactory.cinimex.ru
docker push artifactory.cinimex.ru/vtbmsa_docker/vtbmsa/vtb-cinimex-concentrator:0.2.17-0-3de23dc

Поставка в Банк

После проверки, что кластер Tarantool рабочий, нужно передать его в Банк.

Делаем запись в cahngelog репозитория https://src.cinimex.ru/VTBMSA/vtbmsa-tarantool-cache. Делаем релиз с тэгом версии образа и пушим в Банковский репозиторий в Bitbucket (https://bitbucket.region.vtb.ru/projects/KFT).