Arenadata™ Grid

Версия - v2.3.1-nsud

Кластеризация в Arenadata Grid

2020 Arenadata™

Оглавление

Arenadata Grid обладает расширенными возможностями кластеризации, включая логические группы хранилищ и экземпляров приложений.

Узлы **Arenadata Grid** представляют собой серверные процессы **Tarantool**, выполняющие различные функции. Механизм кластеризации осуществляется с помощью фреймворка **Cartridge**.

Tarantool Cartridge рекомендуется в качестве альтернативы традиционному подходу разработки приложений для **Tarantool**.

В качестве набора средств разработки (SDK) **Tarantool Cartridge** предлагает утилиты и шаблоны, которые помогут:

- С легкостью настроить среду разработки для приложений;
- Подключить необходимые Lua-модули.

Полученный пакет можно установить и запустить на одном сервере или нескольких серверах в качестве одного экземпляра службы или как несколько служб, независимо или в кластере.

Для того, чтобы полностью использовать возможности Tarantool-кластера, необходимо разрабатывать приложения исходя из того, что они должны работать в кластерной среде.

Кроме того, кластер в **Tarantool** предоставляет следующие ключевые преимущества для кластерных приложений:

- Горизонтальная масштабируемость и балансировка нагрузки с помощью встроенного автоматического шардинга;
- Асинхронная репликация;
- Автоматическое восстановление после отказа;
- Централизованное управление кластером через графический интерфейс или API;
- Автоматическая синхронизации настроек;
- Разделение функций экземпляров.

Кластер в **Tarantool Cartridge** может распределять функции экземпляров с помощью встроенных и пользовательских (настраиваемых) кластерных ролей. Во время работы кластера можно включить и отключить роли на лету. Это позволяет по-разному нагружать физические серверы с различным профилем оборудования (например, для ресурсоемких вычислений или для обработки большого количества транзакций).

Основные роли в кластере **Cartridge**:

- **Storage** – экземпляр хранилища данных, который ответственнен за собственно хранение и доступ к данным, хранящимся в памяти (движок *memtx*) и на локальных дисках (движок *vinyl*).
- **Router** – экземпляр приложения в Tarantool. Все роли приложения наследуются от этой роли.

Вышеописанные экземпляры могут объединяться в логические группы *vshard*, обеспечивающие отказоустойчивость и балансировку нагрузки по принципу *1 master + n replicas*. Горизонтальное масштабирование осуществляется автоматическим распределением данных между узлами *vshard*. Данные в таблицах разбиваются на небольшие блоки данных – бакеты. Распределение данных между *vshard* осуществляется путем вычисления из первичного индекса таблицы *ключа шардирования*. Данные с одним и тем же ключом шардирования помещаются в один и тот же *vshard*. Таким образом можно обеспечить локальность данных.

Параметр *Weight* позволяет настроить долю хранимых на конкретном *vshard* бакетов от общего числа бакетов в хранилище.

Экземпляр *master* в шарде позволяет писать и читать данные из таблицы, *replica* – только читать.

Important: Контактная информация службы поддержки – e-mail: info@arenadata.io