|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

Институт информационных технологий (ИИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Распределенные системы управления базами данных»

**Практическое занятие № 2**

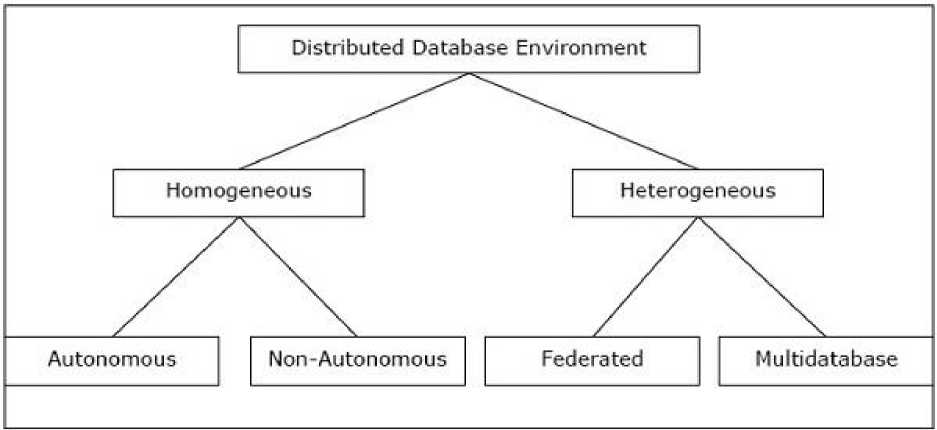
|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИКБО-11-22, Гришин Андрей Валерьевич* | (подпись) | |
| Преподаватель | *Красников Степан Альбертович* | (подпись) | |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_202\_\_г. | |  | |

Москва 2025 г.

**Теоретическое введение**

**Типы распределённых систем**

Распределенные базы данных можно широко классифицировать на однородные и гетерогенные среды распределенных баз данных, каждая из которых имеет дополнительные подразделения, как показано на следующем рисунке:



Однородные распределенные базы данных

Разберём, как работают однородные РБД на примере сайтов (подразумевается некое веб-приложение, установленное на различных узлах). В однородной распределенной базе данных все сайты используют идентичные СУБД и операционные системы. Её свойства —

* На сайтах используется очень похожее программное обеспечение.
* Сайты используют идентичные СУБД или СУБД одного и того же производителя.
* Каждый сайт знает обо всех других сайтах и взаимодействует с другими сайтами для обработки пользовательских запросов.
* Доступ к базе данных осуществляется через единый интерфейс, как если бы это была одна база данных.

Типы однородной распределенной базы данных

Существует два типа однородной распределенной базы данных —

* **Автономный —** каждая база данных независима и функционирует самостоятельно. Они интегрированы управляющим приложением и используют передачу сообщений для обмена обновлениями данных.
* **Неавтономный —** данные распределяются по однородным узлам, а центральная или главная СУБД координирует обновления данных по сайтам.

Гетерогенные распределенные базы данных

В гетерогенной распределенной базе данных разные сайты имеют разные операционные системы, продукты СУБД и модели данных. Его свойства —

* Различные сайты используют разные схемы и программное обеспечение.
* Система может состоять из множества СУБД, таких как реляционная, сетевая, иерархическая или объектно-ориентированная.
* Обработка запросов является сложной из-за разнородных схем.
* Обработка транзакций является сложной из-за различий в программном обеспечении.
* Сайт может не знать о других сайтах, поэтому сотрудничество при обработке пользовательских запросов ограничено.

Типы гетерогенных распределенных баз данных

* **Федеративные** — гетерогенные системы баз данных независимы по своей природе и объединены вместе, так что они функционируют как единая система баз данных.
* **Без федерации** — в системах баз данных используется центральный координационный модуль, через который осуществляется доступ к базам данных.

**Практическая часть:**



Рисунок 1 – Запуск контейнер с первым узлом

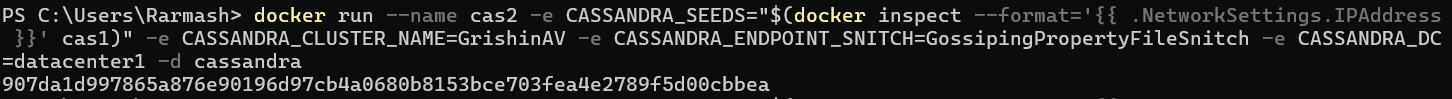


Рисунок 2 – Присоединение к первому узлу следующего узла

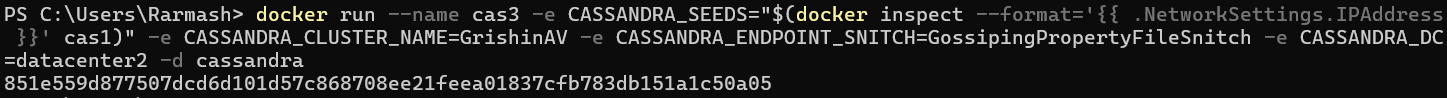


Рисунок 3 – Создание узла в «удалённом» датацентре для образования кластера

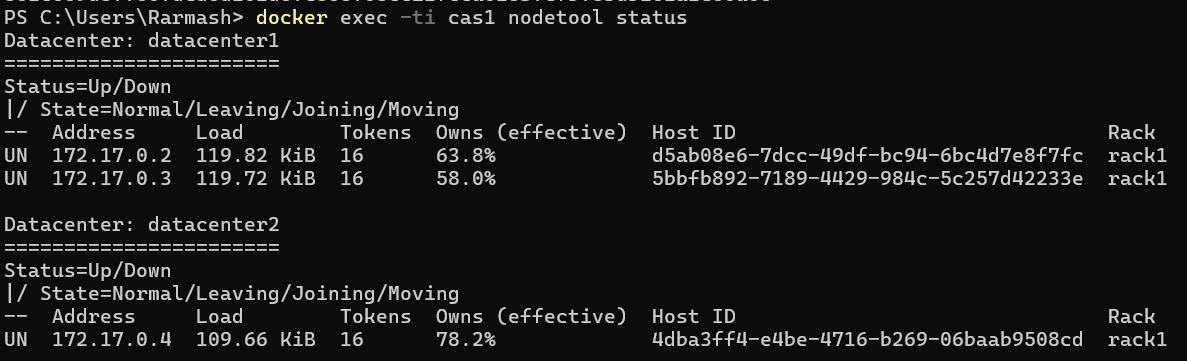


Рисунок 4 – Проверка работоспособности кластера

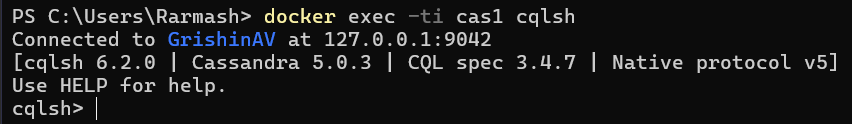


Рисунок 5 - Переход в утилиту cqlsh для выполнения скрипта

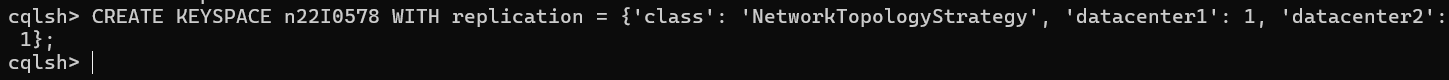


Рисунок 6 – Создание пространства keyspace



Рисунок 7 – Выполнение скрипта для использования keyspace

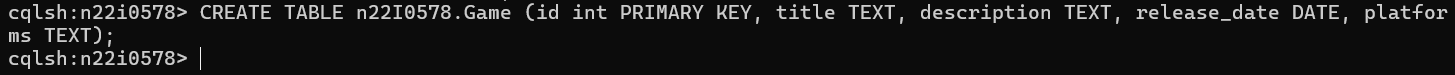


Рисунок 8 – Создание таблицы

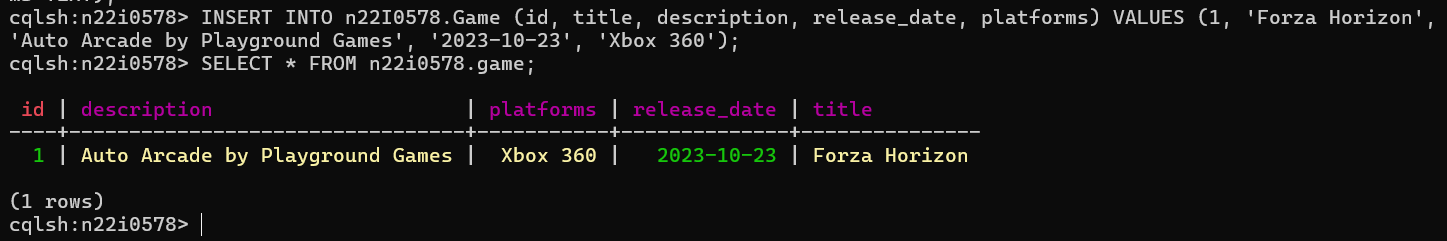


Рисунок 9 – Заполнение ранее созданной таблицы и её вывод

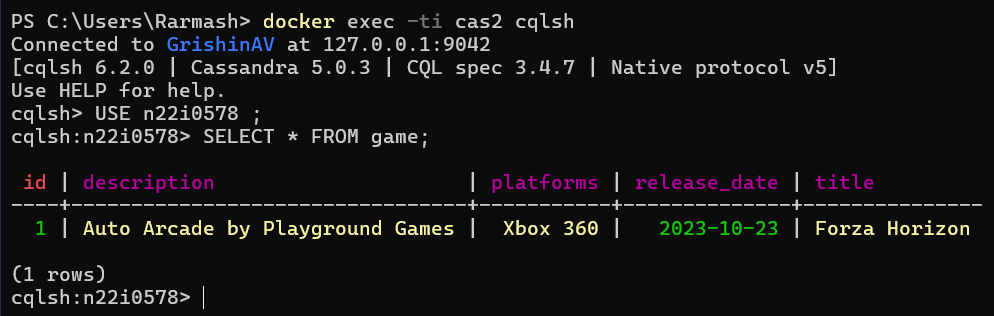


Рисунок 10 – Вывод таблицы со второго узла

Заключение:

В рамках практической работы был создан кластер Apache Cassandra, состоящий из трёх взаимосвязанных узлов. Применение контейнеризации с помощью Docker позволило быстро и удобно развернуть распределённую NoSQL-базу данных. Будучи колоночной NoSQL-системой, Cassandra отлично подходит для обработки больших объёмов информации и для работы в высоконагруженных системах. Её распределённая архитектура обеспечивает автоматическую репликацию данных между узлами, что гарантирует отказоустойчивость и поддержку горизонтального масштабирования. В итоге, развернутый и протестированный кластер, а также изученные команды Docker для управления Cassandra, предоставляют возможность создания надёжных и высокопроизводительных систем.

Контрольные вопросы:

1. Cassandra относится к распределённым **NoSQL** базам данных, а точнее — к **колонно-ориентированным** БД. Это объясняется тем, что Cassandra:

* Не использует реляционную модель данных (таблицы, связи и т.д.).
* Проектирована для горизонтального масштабирования, распределённого хранения данных и высокой доступности.
* Работает по архитектуре peer-to-peer, что позволяет обеспечить отказоустойчивость и распределённое управление данными.

1. Для загрузки (pull) образа Cassandra с Docker Hub используется команда **docker pull cassandra**