ПРАКТИЧЕСКАЯ РАБОТА 9. АДМИНИСТРИРОВАНИЕ СУБД

9.1 Цель и содержание

Цель работы: ознакомиться с администрированием баз данных mongo.

Задачи работы: научиться работать с ролями в БД, научиться производить резервирование баз данных.

9.2 Управление и диагностика

С помощью административного интерфейса, входящего в состав MongoDB, можно выполнять многие административные функции. Список некоторых команд представлен далее:

- help – просмотр справки;
- db.help() – получение справки по методам объекта «db»;
- db.commandHelp(<команда>) – справка по команде, имя которой передается в качестве параметра;
- show dbs – список существующих баз данных;
- show collections – список коллекций текущей базы;
- use <имя БД> – сделать текущей базу данных;
- db.getName() – посмотреть имя текущей БД;
- db – посмотреть имя текущей БД;
- db.version() – вывести версию программы;
- db.getMongo() – получить список текущих подключений.

Для копирования базы данных с одного сервера на другой используются следующие команды:

- db.cloneDatabase(<hostname>) – производит копирование удаленной базы данных в текущую. <hostname> – имя хоста, с которого производится копирование БД;
- db.copyDatabase(<origin>, <destination>, <hostname>) – производит копирование удаленной базы данных с хоста «<hostname>» с именем «<origin>», в базу данных с именем «<destination>»;

- db.dropDatabase() – удаляет текущую базу данных.

Для диагностики состояния базы данных используются следующие команды:

- db.printCollectionStats() – выводит информацию по всем коллекциям БД;
- db.stats() – выводит информацию о базе данных;
- db.serverStatus() – выводит информацию о статусе сервера: информацию о хосте, на котором выполняется процесс «mongod» и информацию о процессе и его настройках;
- db.repairDatabase() – восстанавливает базу данных после аварийного завершения процесса, сбоя питания или в других случаях, когда повреждено содержимое БД. Команда проверяет все данные на испорченность, удаляет найденные некорректные данные и собирает файлы данных. Если сбой произошел на рабочем узле, то нужно запустить «mongod» с параметром «-repair», а после окончания работы перезапустить его в нормальном режиме;
- db.shutdownServer() – остановить сервер базы данных;
- logpath <file> – указывается, в какой файл будет происходить вывод сообщений логирования. По умолчанию, MongoDB выводит сообщения на стандартный вывод (в консоль).

Если MongoDB использует реплики, то получение информации о них возможно с помощью следующих функций: db.getReplicationInfo(), db.printReplicationInfo(), db.printSlaveReplicationInfo().

9.3 Резервирование

Существует несколько способов для создания резервной копии данных, находящихся в базе данных mongo. Наилучшим является способ, который при резервировании не вызывает простоя (блокировки) сервиса. Таким способом является использование утилиты «mongodump.exe», входящей всостав MongoDB.

Для восстановления резервной копии, полученной с использованием «mongodump.exe», используется утилита «mongorestore.exe», которая также входит в состав MongoDB.

Утилита «mongodump.exe» имеет следующие параметры запуска:

--help – вывод справки;

--verbose, -v – степень детализации; выводить больше информации (например, для большей детализации необходимо использовать -vvvvv);

--host – хост mongo;

--port – порт сервера, также возможно использовать – hosthostname:port;

--ipv6 – включить поддержку IPv6 (по умолчанию отключена);

--username, -u – имя пользователя;

--password, -p – пароль;

--dbpath – прямой доступ к файлам базы mongo по указанному пути, вместо подключения к серверу «mongod.exe». Для использования необходима блокировка директории с данными;

--db, -d – указывает базу данных;

--collection, -c – указывает используемые коллекции;

--directoryperdb – если задан аргумент «--dbpath», то каждая БД находится в отдельной директории;

--out, -o – директория вывода;

--query, -q – запрос JSON для ограничения документов, попадающих в вывод «mongodump.exe» (ограничение дампа документов).[17]

Утилита mongodump.exe может подключаться к указанному хосту, для дампа БД. Также можно задать базу данных, резервную копию которой необходимо сделать, либо её коллекцию (или коллекции). Если база данных не указывается, то подразумеваются все базы данных. Если директория, в которую должен происходить вывод данных, не указана (параметр «--out»),то вывод произойдет в поддиректорию «dump», директории запуска утилиты.Результатом работы утилиты является дамп базы/баз данных, представляющий собой директории с именами БД, хранящие файлы в формате BSON.

Утилита «mongorestore.exe» умеет следующие параметры запуска:

--dbpath – прямой доступ к файлам базы mongo по указанному пути, вместо подключения к серверу «mongod.exe». Необходима блокировка директории с данными, поэтому не может быть использована, если mongo-сервер использует базу по указанному пути;

--directoryperdb – если задан аргумент dbpath, то каждая БД находится в отдельной директории;

--objcheck – проверка объектов на валидность перед вставкой;

--filter <json> – использование фильтра перед вставкой;

--drop – удалить каждую коллекцию из целевой БД перед восстановлением;

--noIndexRestore – не восстанавливать индекс.

Для дампа либо восстановления базы данных на локальной машине достаточно указать базу данных и директорию/файл для дампа/восстановления.[18]

Например, для создания резервной копии всех баз данных на локальном компьютере в директорию «dump», утилита вызывается без параметров:

```
mongodump
```

Для создания резервной копии базы данных «test» в папке «d:\mongodb\backup\» используются следующие параметры запуска:

```
mongodump --db test --out d:\mongodb\backup\
```

Утилиту «mongorestore.exe» также можно запускать без параметров,при условии, что дамп данных находится в папке «dump» в директории утилиты. Для восстановления одной базы данных, либо коллекций, утилита вызывается с параметрами аналогичными параметрам «mongodump.exe».

---

[17] Mongodump – MongoDBManual [Электронный ресурс]. – Режим доступа: http://docs.mongodb.org/manual/reference/program/mongodump/

[18] Mongorestore – MongoDBManual [Электронный ресурс]. – Режим доступа: http://docs.mongodb.org/manual/reference/program/mongorestore/

Иногда требуется произвести блокировку базы данных для прямого доступа к файлам. В этом случае необходимо заблокировать БД командой

```
db.fsyncUnlock()

или
use admin
db.runCommand({fsync: 1,lock: 1})
```

Для разблокирования БД используется команда

```
db.fsyncUnlock()
```

Примечание.
Обратите внимание на то, что в режиме блокировки все транзакции на запись остановлены, и вызов любой функции, вносящей изменения в БД, приведет к зависанию программы, инициирующей вызов до тех пор, пока блокировка с БД не будет снята.

При использовании репликации в MongoDB целесообразно делать резервные копии на рабочих серверах, а не на главном, т.к. в таком случае снижение производительности будет минимальным.

9.4 Работа с пользователями в MongoDB

До версии 2.4 в MongoDB учетные записи пользователей имели только следующие привилегии:

- только чтение базы данных;
- чтение и запись.

Начиная с версии 2.4, в MongoDB используются расширенные роли пользователей. Теперь роли пользователя не ограничены только чтением,либо записью всех баз данных, появилась более гибкая настройка. Одному пользователю возможно назначать различные роли для каждой базы данных. Ограничения для отдельных коллекций либо документов по-прежнему не доступны.

В MongoDB роли только предоставляют доступ, но никогда не ограничивают его. Именно поэтому пользователю в MongoDB нельзя запретить доступ к коллекции либо документу.

MongoDB поддерживает модель работы с пользователями, которая присутствовала в версии 2.2 (и более ранних версиях), в которой для добавления пользователя в базу данных используется команда «addUser» применительно к объекту БД. Синтаксис команды:

db.addUser(<имя>, <пароль>, {<только чтение>: true | false}),

где:

<имя> – имя пользователя;

<пароль> – пароль пользователя;

<только чтение> – ***true*** – только чтение базы данных, ***false*** – чтение и запись.

Данная команда добавляет пользователя, устанавливая ему режим доступа ко всей базе данных.

Для добавления пользователя с назначением ему роли используется команда «addUser», принимающая в качестве параметра сформированный документ «system.users». Формат документа:

```
{
   user: "<username>",
   pwd: "<hash>",
   roles: []
}
```

В документе поле «user» должно содержать имя пользователя, «pwd» – пароль, а массив «roles» – роли пользователя. Поле «pwd», после сохранения документа, содержит хэш пароля пользователя.

Например, команда для добавления пользователя с привилегиями на чтение и запись применительно к текущей БД имеет вид:

```
db.addUser( { user: "ivan", pwd: "pass", roles: [ "readWrite" ] } )
```

В случае, если необходимо указать пользователя, описанного в другой БД, документ принимает следующий формат:

```
{
   user: "<username>",
   userSource: "<database>",
   roles: []
}
```

Поле «userSource» указывает базу данных, содержащую описание привилегий пользователя. Поля «pwd» и «userSource» не могут содержаться в одном документе.

Если необходимо для пользователя указать список привилегий для других баз данных, то они перечисляются во вложенном документе «otherDBRoles»:

```
{
   user: "<username>",
   userSource: "<database>",
   otherDBRoles: {
      <database0> : [],
      <database1> : []
   },
   roles: []
}
```

Документ «otherDBRoles» содержит список имен баз данных с указанием ролей для каждой, в нем <database0> – имя базы данных, для которой далее указаны роли и список ролей в массиве, <database1> – имя следующей БД, и т.д.

Следующий пример описывает пользователя с именем «admin» ролью «clusterAdmin» и дополнительными ролями «read» и «dbadmin» для баз данных «config» и «records»:

```
{
  user: "admin", userSource:
  "$external", roles: [
  "clusterAdmin"],
  otherDBRoles:
  {
    config: [ "read" ],
    records: [ "dbadmin" ]
  }
}
```

Для добавления пользователя с описанной выше ролью используется запрос:

```
db.addUser({ user: "admin", userSource: "$external", roles: [ "clusterAdmin"],
otherDBRoles: { config: [ "read" ], records: [ "dbadmin" ]
}})
```

В MongoDB могут быть использованы различные роли.

Пользовательские роли (применительно к текущей БД):

- read. Дает возможность пользователю получить данные из любой коллекции базы данных. Роль дает возможность пользоваться следующими функциями: «find», «aggregate», «checkShardingIndex», «collStats», «count», «dataSize», «dbHash», «dbStats», «distinct», «mapReduce» (только вывод в консоль) и некоторыми другими;
- readWrite. Дает возможность пользователю производить любые операции чтения/записи.

Роли администрирования баз данных (применительно к текущей БД):

- dbAdmin. Предоставляет возможность выполнять следующий набор административных операций в рамках БД: «clean», «collMod», «collStats», «create», «db.createCollection», «dbStats», «drop», «dropIndexes», «ensureIndex», «indexStats», «reIndex», «renameCollection» и некоторые другие. Только dbAdmin может получать данные из коллекции «system.profile»;

- userAdmin. Позволяет пользователям читать и записывать данные в коллекцию «system.users» любой базы данных. Может изменять разрешения для существующих пользователей и создавать новые.

Административные роли: clusterAdmin. Предоставляет возможность администрировать кластер баз данных, дает возможность работать с репликами и шардами. Данная роль не предоставляет доступа к локальным конфигурациям баз данных.

Привилегии для любых баз данных:

- readAnyDatabase. Аналог «read», применительно к любой БД;
- readWriteAnyDatabase. Аналог «readWrite», применительно к любой БД;
- userAdminAnyDatabase. Аналог «userAdmin», применительно к любой БД;
- dbAdminAnyDatabase. Аналог «dbAdmin», применительно к любой БД.

Просмотр списка всех пользователей осуществляется с помощью команды «show users».

Удаление пользователя – командой «db.removeUser(<имя пользователя>)».

9.5 Методика и порядок выполнения работы

Индивидуальное задание:

1. Получите диагностическую информацию о вашей базе данных и содержащихся в ней коллекциях.
2. Создайте резервную копию данных вашей БД.
3. Восстановите базу данных из резервной копии.
4. Создайте для базы данных несколько пользователей, имеющих различные роли.