NOSQL

PROF. ME.
HÉLIO
ESPERIDIÃO

O QUE É NOSQL?

NoSQL é um termo usado para uma estrutura de banco de dados diferente, na qual estamos acostumados a utilizar, os não relacionais

maior diferencial entre esse tipo de banco e os tradicionais são a velocidade e alta escalabilidade.

Hoje, temos 5 modelos diferentes, sendo eles:

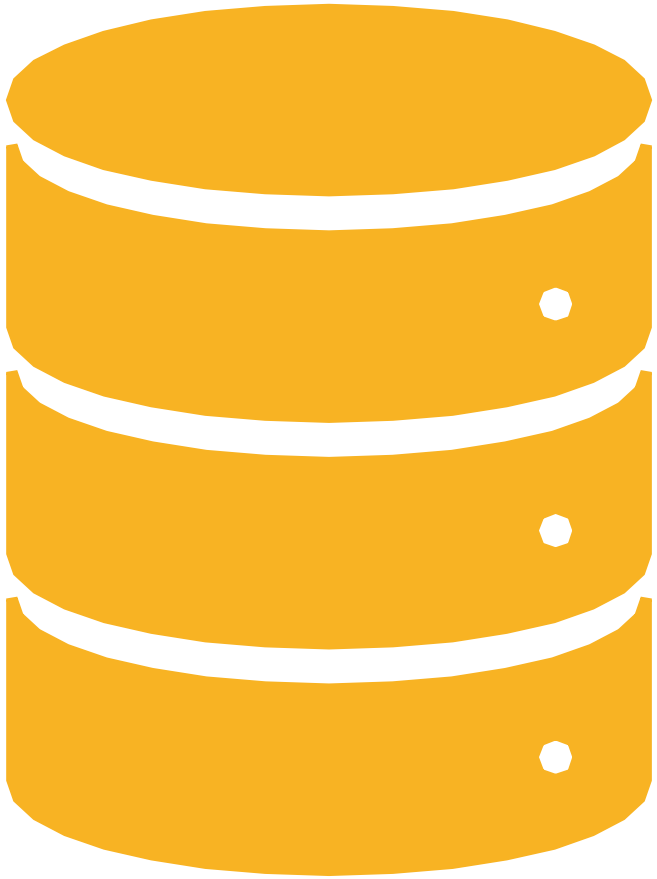
MODELOS

CHAVE-VALOR

DOCUMENTO

CHAVE-VALOR

- Esse modelo é o mais simples entre os bancos de dados NoSQL. Sua estrutura é muito semelhante ao Map do Java.
  - Redis
  - DyanamoDB
  - Scalaris
  - Project Voldemort



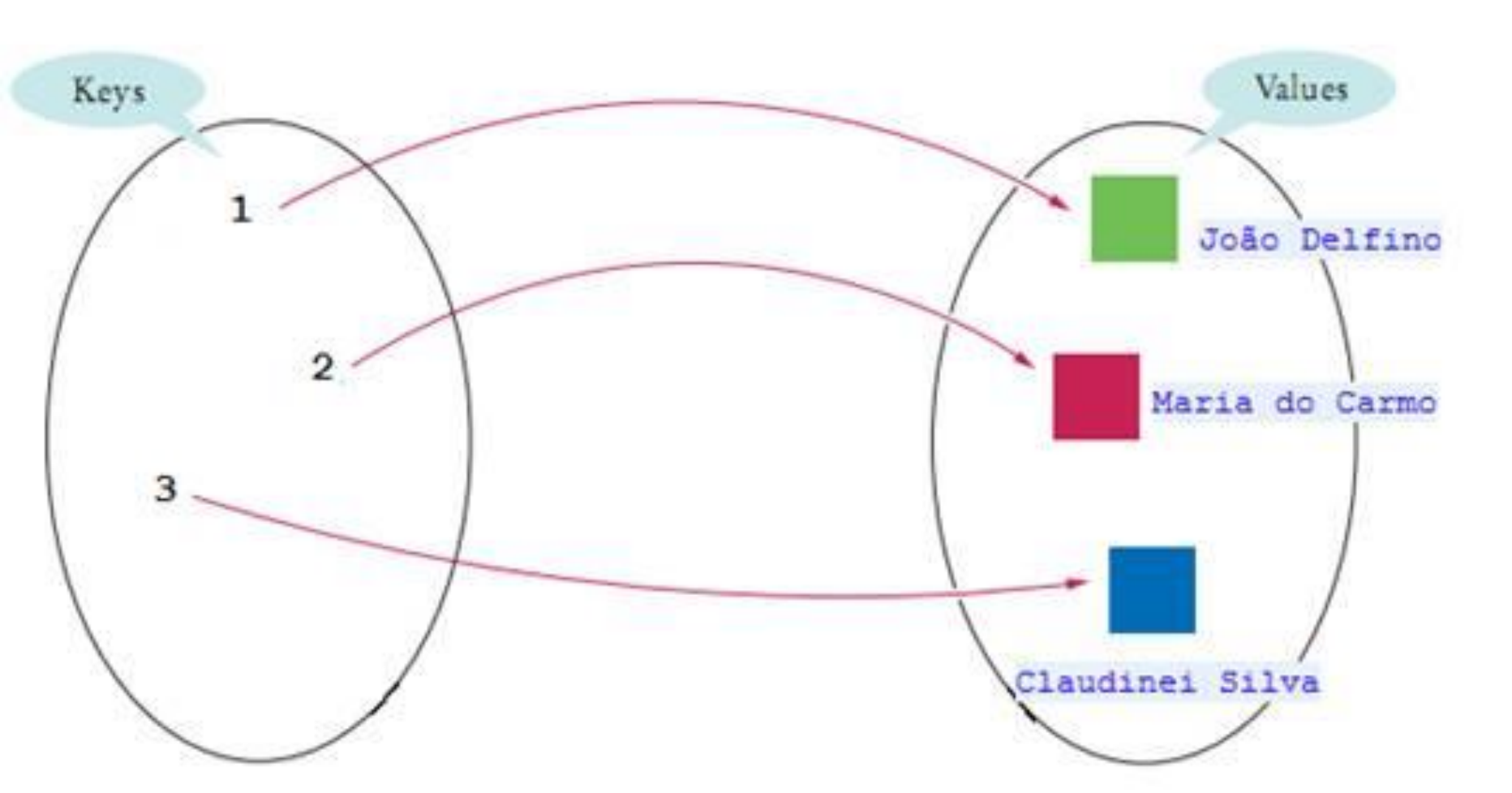
MAPA JAVA

Os objetos “Map” confiam seus dados em um algoritmo hash (hash code).

Esse algoritmo transforma uma grande quantidade de dados em uma pequena quantidade de informações, sendo que o mecanismo de busca se baseia na construção de índices.

Um exemplo prático pode ser usado como uma lista telefônica onde a letra seria o índice a ser procurado, para conseguir achar mais fácil o nome desejado.

MAPA


Keys
Values
1
2
3
João Delfino
Maria do Carmo
Claudinei Silva

EXEMPLO

```
public static void main(String[] args) {
    Map map = new HashMap();
    //Adding elements to map
    map.put(1, "ana");
    map.put(5, "Maria");
    map.put(2, "Paula");
    map.put(6, "Patricia");
    //Converting to Set so that we can traverse
    Set set = map.entrySet();
    Iterator itr = set.iterator();
    //Converting to Map.Entry so that we can get key and value separately
    while (itr.hasNext()) {
        Map.Entry entry = (Map.Entry) itr.next();
        System.out.println(entry.getKey() + " " + entry.getValue());
    }
}
```

DOCUMENTO

Talvez seja o mais conhecido por conta da popularidade do MongoDB e, como consequência, acaba sendo porta de entrada para quem quer fugir dos modelos tradicionais.

Seu diferencial é a possibilidade de armazenar dados semi-estruturados, ou seja, você não precisa ter um schema pré-definido

EXEMPLOS

- SimpleDB
- CouchDB
- MongoDB
- Riak

MONGODB

- MongoDB é uma nova ideia de banco de dados Orientado a Documentos.

BANCO DE DADOS ORIENTADO A DOCUMENTOS

Tem como característica conter todas as informações importantes em um único documento.

É livre de esquemas, possuir identificadores únicos universais (UUID).

Possibilitar a consulta de documentos através de métodos avançados de agrupamento e filtragem (MapReduce).

NOSQL

Também são chamados de Bancos NoSQL (Not Only SQL).

NoSQL é devido à ausência do SQL.

Alguns chegaram a defender o termo NoREL (Not Relational), mas diferente do anterior este não foi muito aceito.

De forma resumida esse tipo de Banco de Dados não traz consigo as ideias do modelo relacional e nem a linguagem SQL.

CARACTERÍSTICAS DO MONGODB

- Código-fonte aberto licenciado pela GNU AGPL (Affero General Public License) versão 3.0
- possuir alta performance, não possuir esquemas, ser escrito em C++
- Multiplataforma e ser formado por um conjunto de aplicativos JSON
- Diversas linguagens e plataforma já possuem drivers para o MongoDB, entre elas destacam-se: C, C#, C++, Haskell, Java, JavaScript, Perl, PHP, Python, Ruby e Scala
- Globo.com, SourceForge, FourSquare, MailBox (serviço de e-mail do Dropbox), LinkedIn, SAP, MTV, Pearson Education

JSON - JAVASCRIPT OBJECT NOTATION

É uma formatação leve de troca de dados.

Para seres humanos, é fácil de ler e escrever.

Para máquinas, é fácil de interpretar e gerar.

É baseado em um subconjunto da linguagem de programação JavaScript, Standard ECMA-262 3a Edição -Dezembro - 1999.

JSON

JSON é em formato texto e completamente independente de linguagem

Formato ideal de troca de dados

é um formato compacto, de padrão aberto independente, de troca de dados simples e rápida (parsing) entre sistemas

ESTRUTURA

Uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um *object*, record, struct, dicionário, hash table, keyed list, ou arrays associativas.

Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como uma *array*, vetor, lista ou sequência.

EXEMPLO DE JSON MONGODB

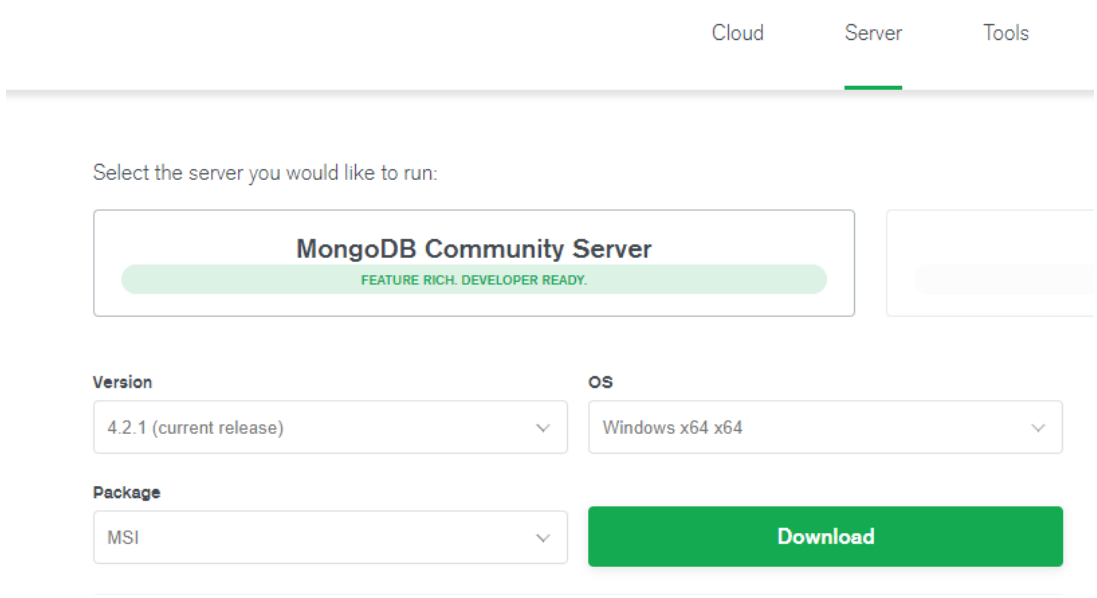
- { "_id" : { "$oid" : "5dc1d7b8c971ab0a28326098" }, "cpf" : "33333333390", "rg" : "405558592", "nome" : "Hélio", "Curso" : "Administração" }

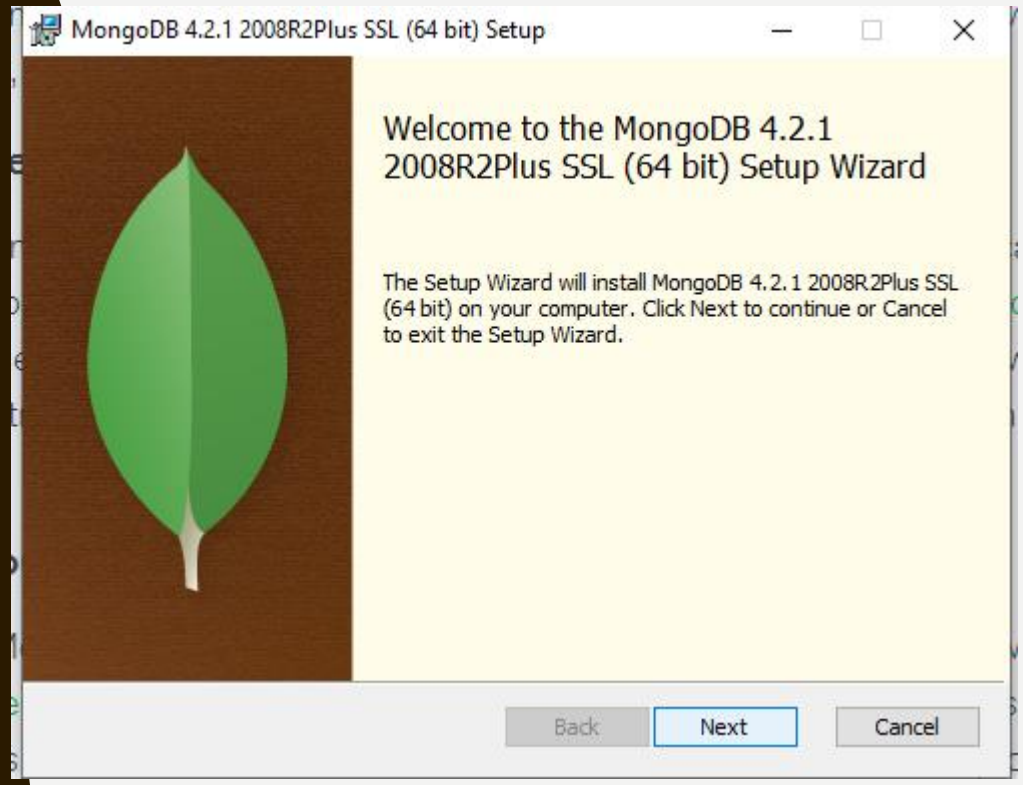
- { "_id" : { "$oid" : "5dc1e7124f01312cd4b4e166" }, "cpf" : "12345678911111", "rg" : "456456", "nome" : "Andreia", "Curso" : "Engenharia" }

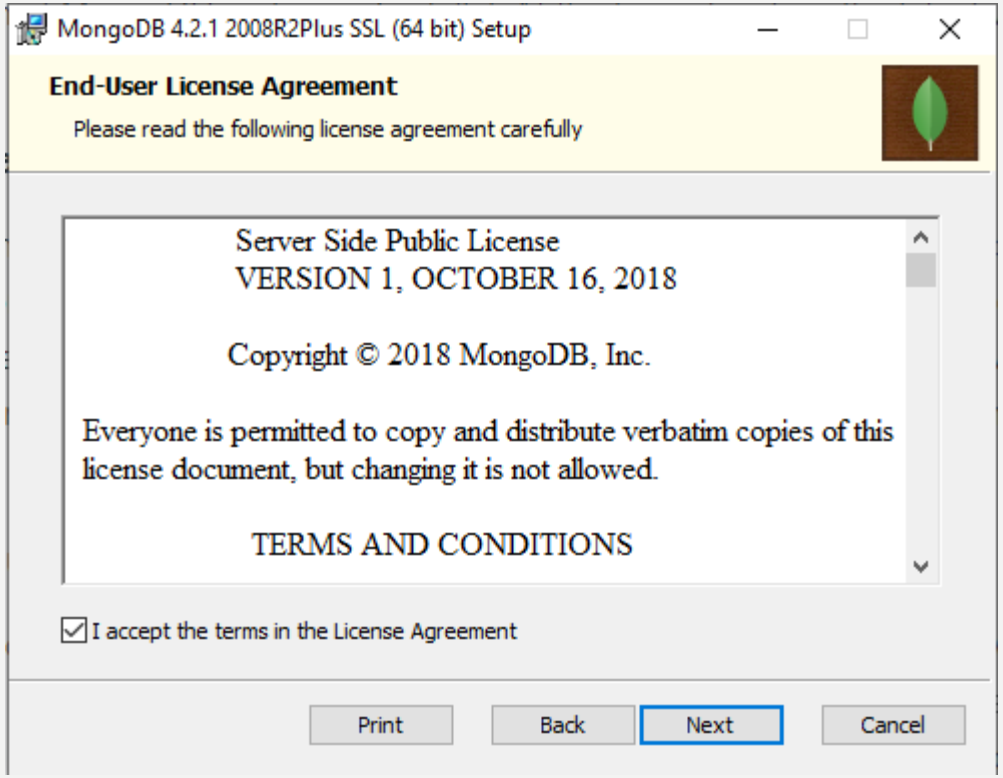
INSTALANDO O MONGODB

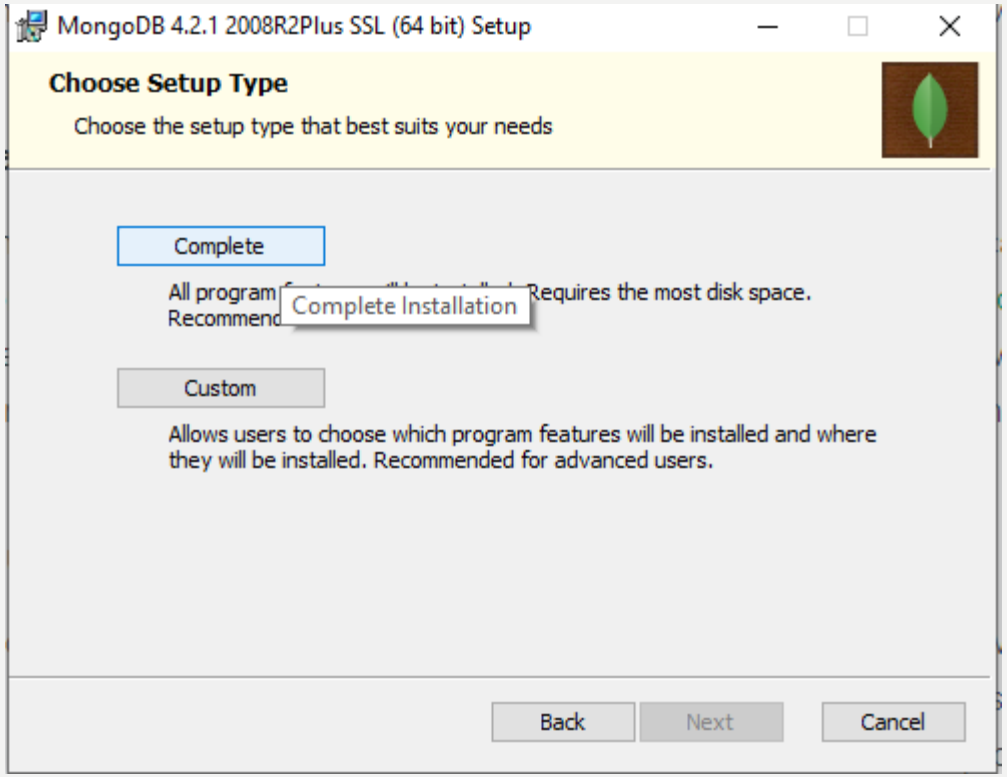
- Para instalar o MongoDB devemos primeiramente baixa-lo, escolhendo uma versão de sistema operacional.

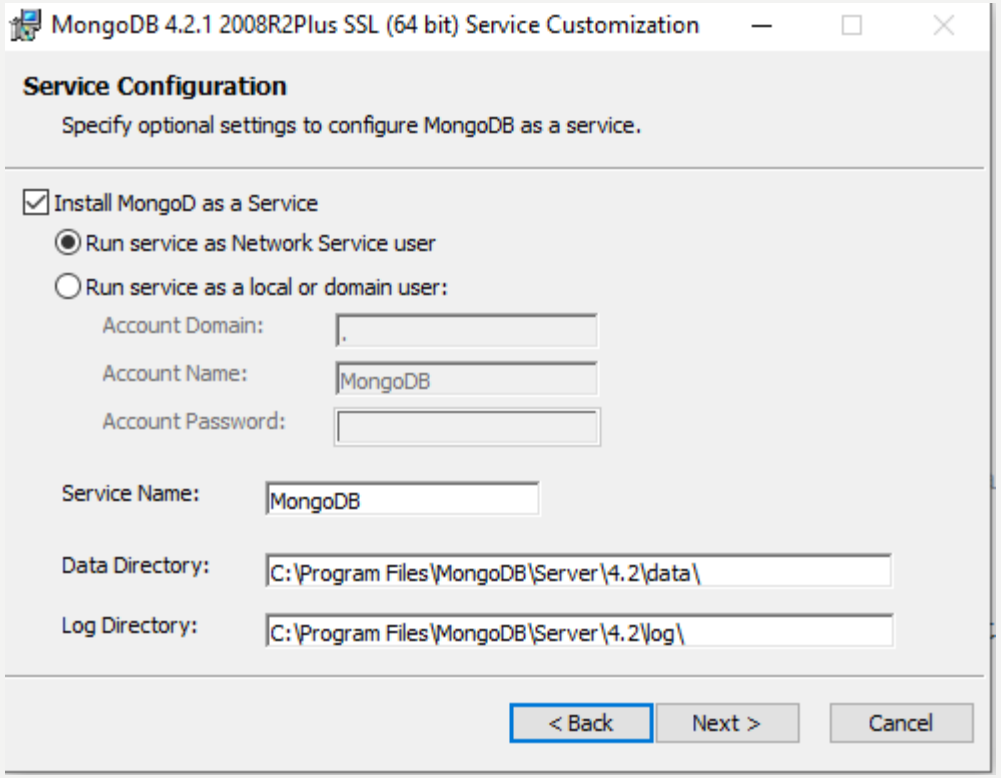
- https://www.mongodb.com/download-center

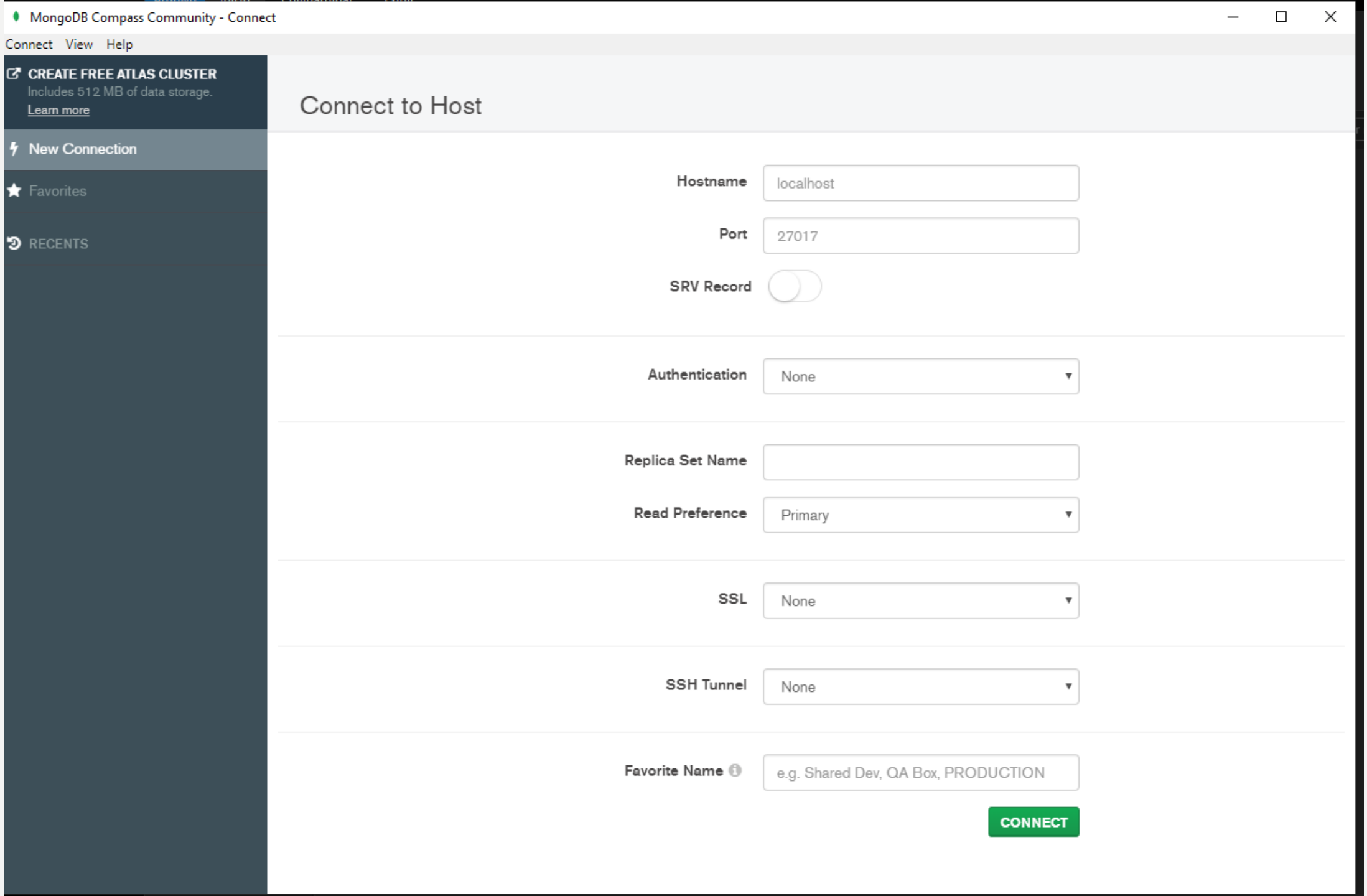



MongoDB 4.2.1 2008R2Plus SSL (64 bit) Setup
Welcome to the MongoDB 4.2.1 2008R2Plus SSL (64 bit) Setup Wizard
The Setup Wizard will install MongoDB 4.2.1 2008R2Plus SSL (64 bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
Back
Next
Cancel

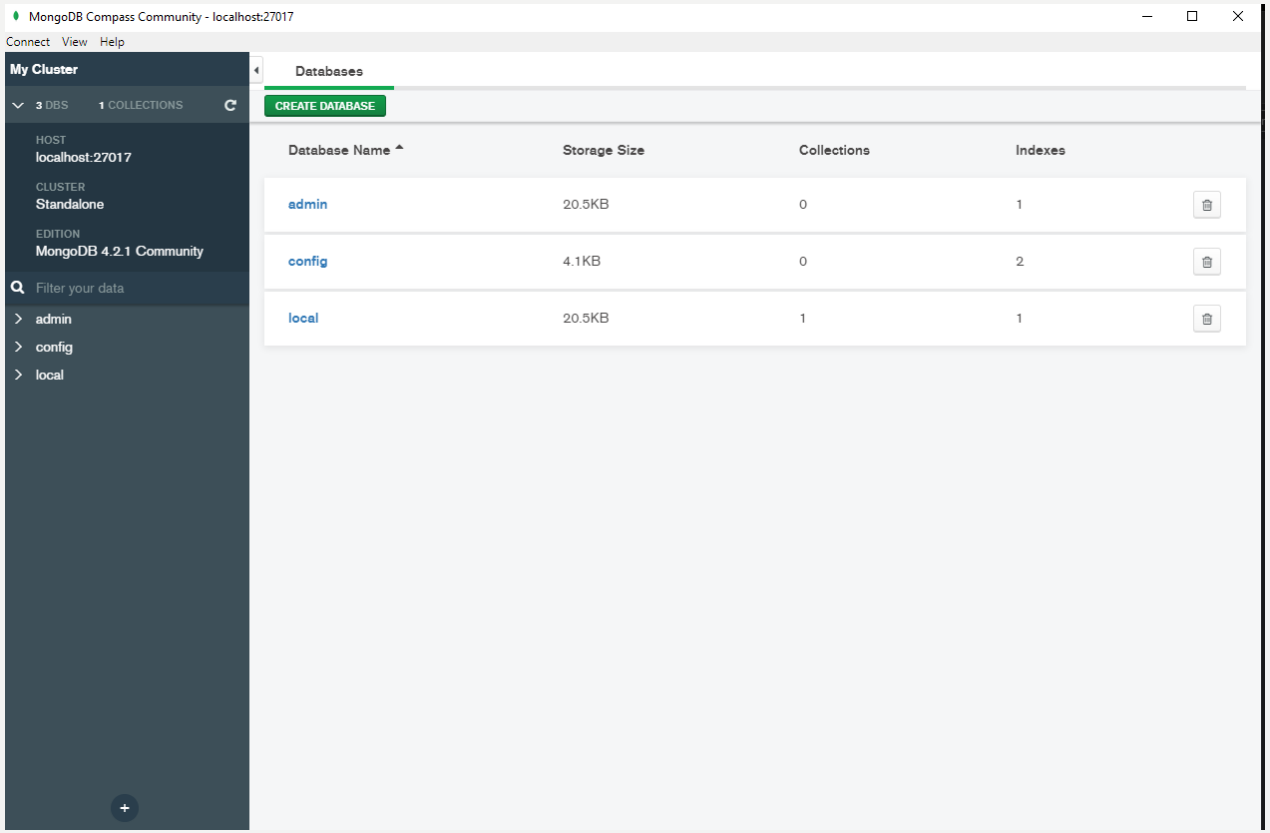

MongoDB 4.2.1 2008R2Plus SSL (64 bit) Setup
End-User License Agreement
Please read the following license agreement carefully
Server Side Public License
VERSION 1, OCTOBER 16, 2018
Copyright © 2018 MongoDB, Inc.
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
TERMS AND CONDITIONS
I accept the terms in the License Agreement
Print
Back
Next
Cancel

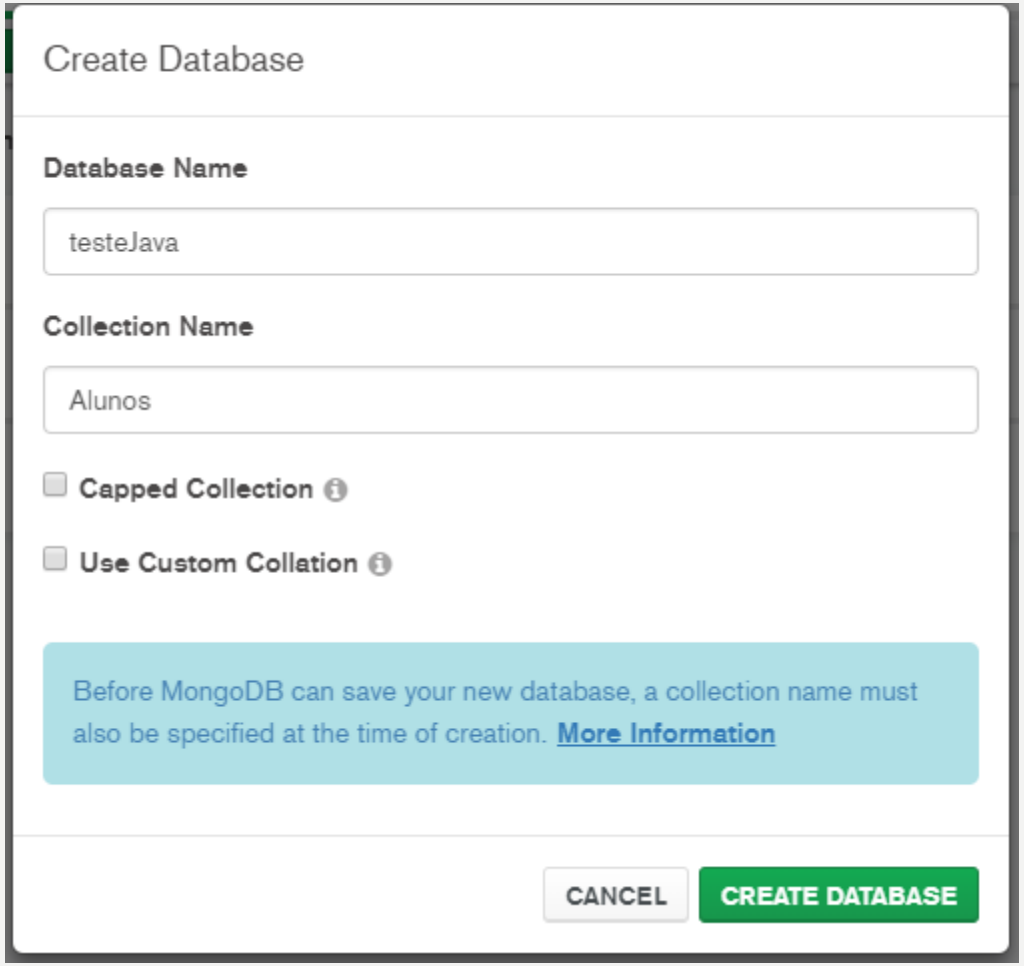

MongoDB 4.2.1 2008R2Plus SSL (64 bit) Setup
Choose Setup Type
Choose the setup type that best suits your needs
Complete
All program
Requires the most disk space.
Recommend
Complete Installation
Custom
Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.
Back
Next
Cancel

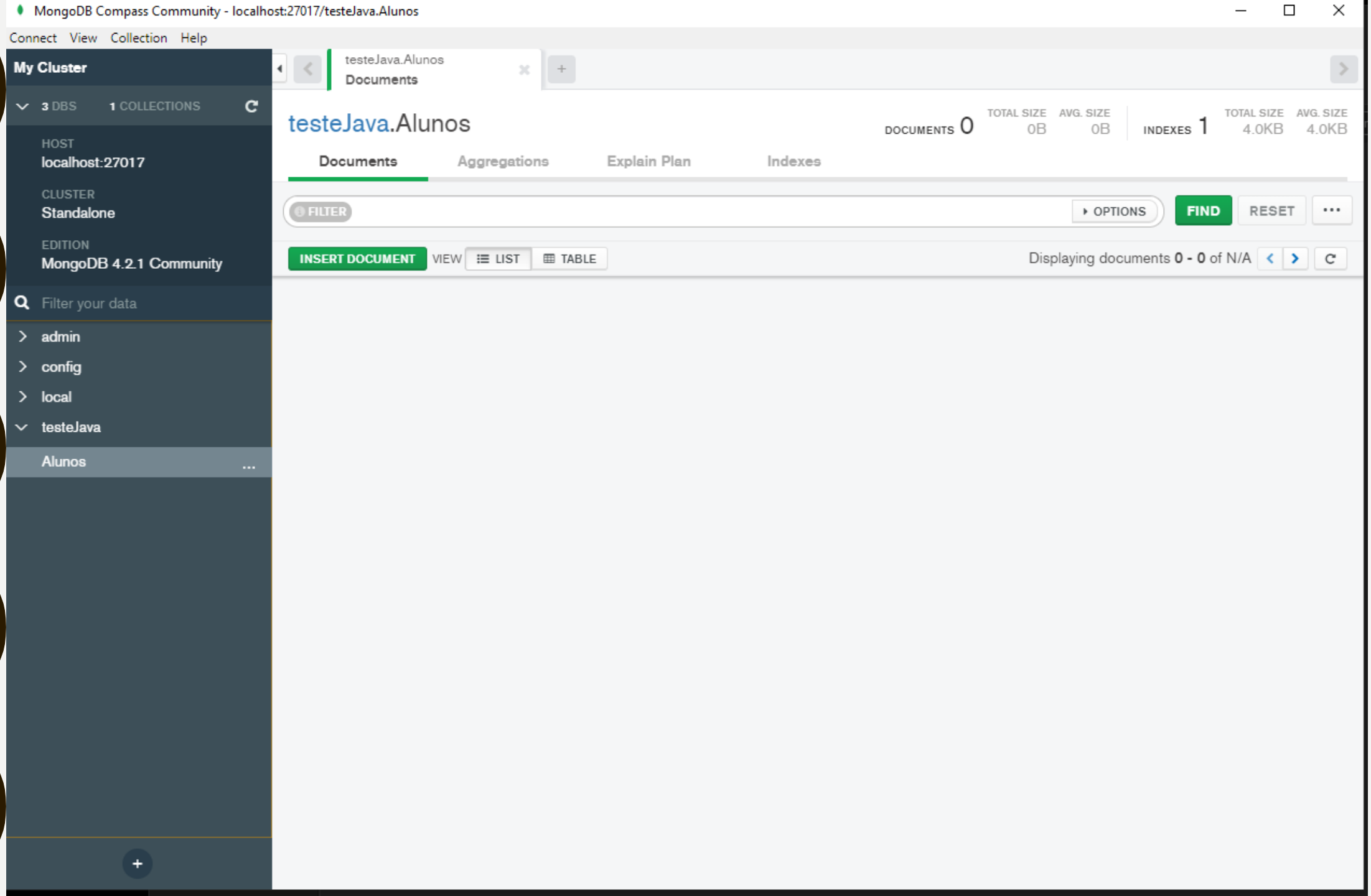

MongoDB 4.2.1 2008R2Plus SSL (64 bit) Service Customization
Service Configuration
Specify optional settings to configure MongoDB as a service.
Install MongoD as a Service
Run service as Network Service user
Run service as a local or domain user:
Account Domain:
.
Account Name:
MongoDB
Account Password:
Service Name:
MongoDB
Data Directory:
C:\Program Files\MongoDB\Server\4.2\data\
Log Directory:
C:\Program Files\MongoDB\Server\4.2\log\
< Back
Next >
Cancel


MongoDB Compass Community - Connect
Connect View Help
CREATE FREE ATLAS CLUSTER
Includes 512 MB of data storage.
Learn more
New Connection
Favorites
RECENTS
Connect to Host
Hostname
localhost
Port
27017
SRV Record
Authentication
None
Replica Set Name
Read Preference
Primary
SSL
None
SSH Tunnel
None
Favorite Name
e.g. Shared Dev, QA Box, PRODUCTION
CONNECT

CREATE DATABASE




Create Database
Database Name
testeJava
Collection Name
Alunos
Capped Collection
Use Custom Collation
Before MongoDB can save your new database, a collection name must also be specified at the time of creation. More Information
CANCEL
CREATE DATABASE

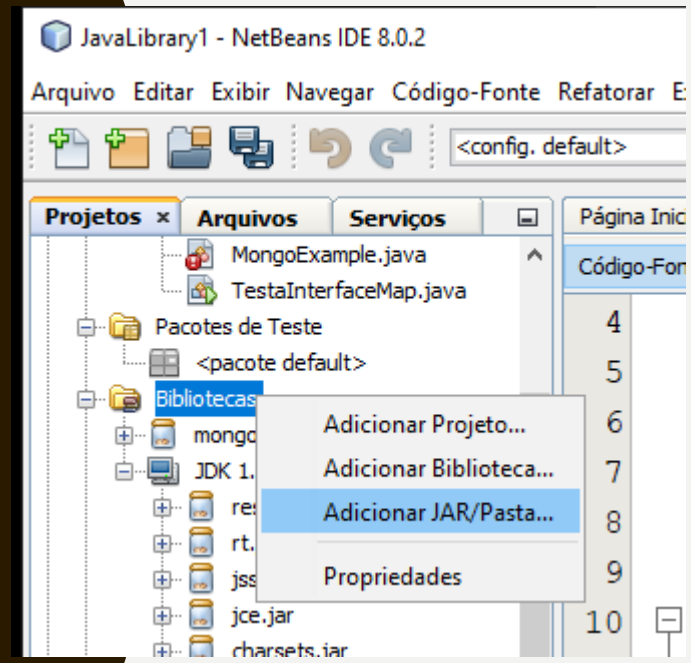

MongoDB Compass Community - localhost:27017/testeJava.Alunos
Connect View Collection Help
My Cluster
3 DBS 1 COLLECTIONS
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 4.2.1 Community
Filter your data
admin
config
local
testeJava
Alunos
testeJava.Alunos
Documents
testeJava.Alunos
DOCUMENTS 0
TOTAL SIZE 0B
AVG. SIZE 0B
INDEXES 1
TOTAL SIZE 4.0KB
AVG. SIZE 4.0KB
Documents
Aggregations
Explain Plan
Indexes
FILTER
OPTIONS
FIND
RESET
INSERT DOCUMENT
VIEW
LIST
TABLE
Displaying documents 0 - 0 of N/A

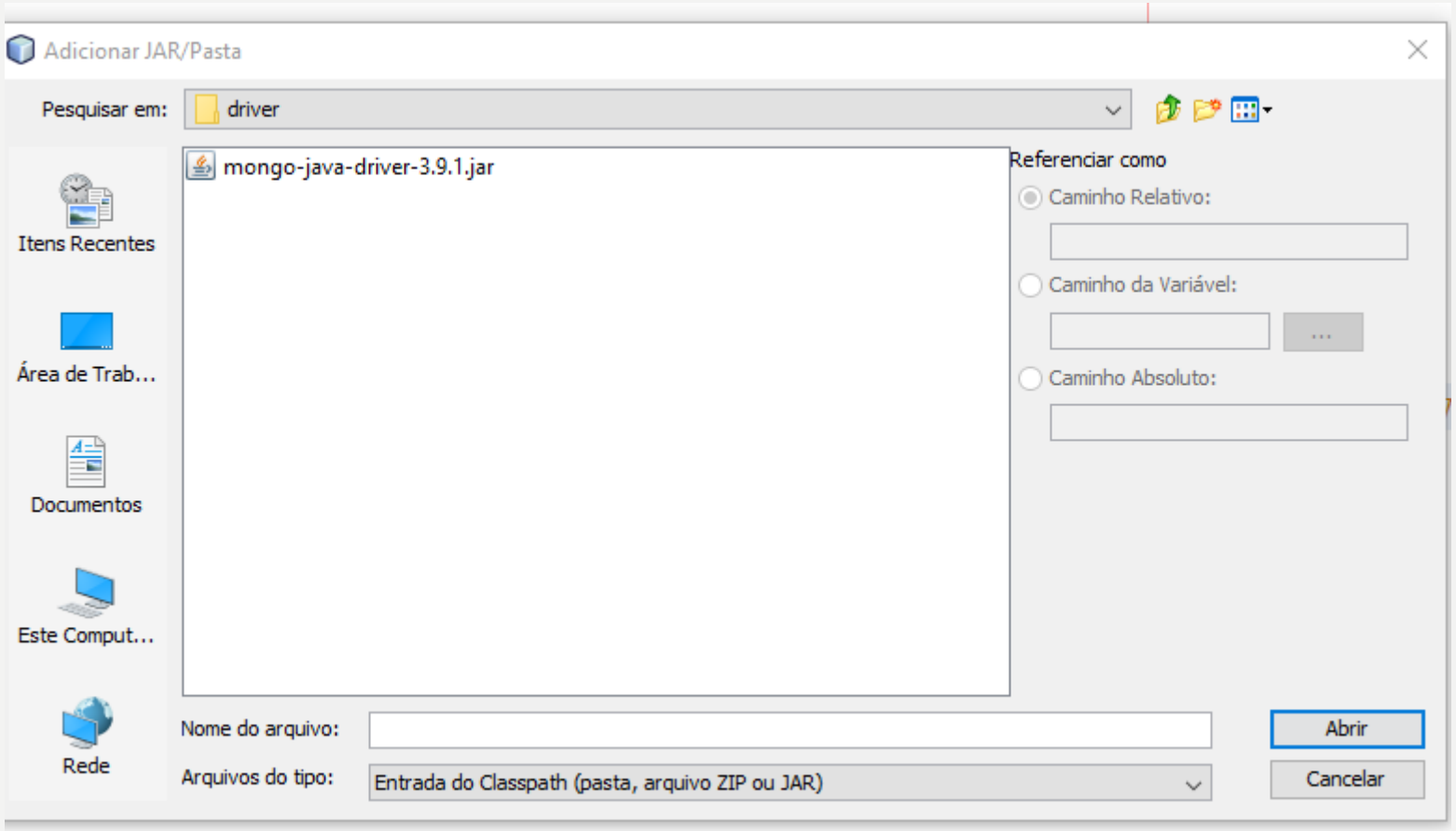
INSERINDO DADOS NA COLEÇÃO

DRIVER

- mongo-java-driver-3.9.1.jar
- http://central.maven.org/maven2/org/mongodb/mongo-java-driver/

ADD O DRIVER NO NETBEANS





LISTAR DOCUMENTOS

```
public String ListarDocumentos(String nomeColecao) {
    Logger mongoLogger = Logger.getLogger("org.mongodb.driver");
    mongoLogger.setLevel(Level.SEVERE);
    this.mongoDB = new MongoClient("localhost", 27017);
    this.db = mongoDB.getDatabase("testeJava");
    MongoCollection<Document> collection = db.getCollection(nomeColecao);
    String resposta = "";
    FindIterable<Document> cursor = collection.find();
    for (Document doc : cursor) {
        resposta+=doc.toJson();
        //System.out.println(doc.getString("cpf"));
        //System.out.println(doc.toJson());
    }
    return resposta;
}
```

INSERIRDOCUMENTO

```
public void inserirDocumento(String nomeColecao) {
    Logger mongoLogger = Logger.getLogger("org.mongodb.driver");
    mongoLogger.setLevel(Level.SEVERE);
    this.mongoDB = new MongoClient("localhost", 27017);
    this.db = mongoDB.getDatabase("testeJava");
    MongoCollection<Document> collection = db.getCollection(nomeColecao);
    Document doc = new Document("cpf", "1234567891111").append("rg", "456456").append("nome", "Andreia").append("Curso", "Engenharia");
    collection.insertOne(doc);
}
```

LOCALIZARDOCUMENTO

```
public void localizarDocumento(String nome, String nomeColecao) {
    Logger mongoLogger = Logger.getLogger("org.mongodb.driver");
    mongoLogger.setLevel(Level.SEVERE);
    this.mongoDB = new MongoClient("localhost", 27017);
    this.db = mongoDB.getDatabase("testeJava");
    MongoCollection<Document> collection = db.getCollection(nomeColecao);
    try (MongoCursor<Document> cursor = collection.find(eq("nome", nome)).iterator()) {
        while (cursor.hasNext()) {
            Document atual = cursor.next();
            System.out.println(atual.toJson());
        }
    }
}
```

ALTERAR DOCUMENTO

```
public void alteraAluno(String nomeAntigo, String nomeNovo,String nomeColecao) {
    Logger mongoLogger = Logger.getLogger("org.mongodb.driver");
    mongoLogger.setLevel(Level.SEVERE);
    this.mongoDB = new MongoClient("localhost", 27017);
    this.db = mongoDB.getDatabase("testeJava");
    MongoCollection<Document> collection = db.getCollection(nomeColecao);
    collection.updateOne(eq("nome", nomeAntigo), new Document("$set", new Document("nome", nomeNovo)));
}
```

REMOVER DOCUMENTO

```
public void removerAluno(String nome,String nomeColecao){
        Logger mongoLogger = Logger.getLogger("org.mongodb.driver");
        mongoLogger.setLevel(Level.SEVERE);
        this.mongoDB = new MongoClient("localhost", 27017);
        this.db = mongoDB.getDatabase("testeJava");
        MongoCollection<Document> collection = db.getCollection(nomeColecao);
          collection.deleteOne(eq("nome", nome));
}
```

TESTANDO

```
public static void main(String[] args) {
    MongoExample mongo = new MongoExample();

    mongo.inserirDocumento("Alunos");
    mongo.localizarDocumento("Andreia", "Alunos");
    mongo.alteraAluno("Andreia", "Joada", "Alunos");
    mongo.removerAluno("Joada", "Alunos");
    System.out.println(mongo.ListarDocumentos("Alunos"));

}
```