

COMPSCI 326

Web Programming

19 – Heroku, Node, NoSQL, App

Lecture Material: <https://bityl.co/Br7D>

Course Material On Github: <https://bityl.co/BmsC>

Today

- NoSQL Databases
- MongoDB CRUD Operations
- Setup MongoDB on Atlas
- CRUD with Express, MongoDB, and Heroku

NoSQL

Broadly speaking, there are two kinds of databases: SQL and "NoSQL". NoSQL databases are diverse.

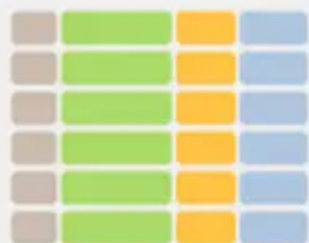
Some, known as **key-value stores**, are little more than CRUD databases, with calls like **put**, **get**, **delete** (basically, create/update, read, and delete), and very limited query ability.

Others, like MongoDB, are "document oriented", and directly let you manage JSON - we'll talk about MongoDB today.

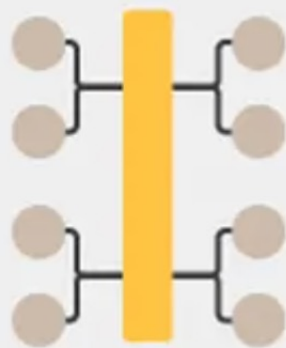


SQL Database

Relational

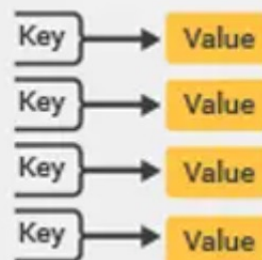


Analiticals (OLAP)



Non-SQL Database

Key-Value



Column-Family



Graph



Document



Why NoSQL?

- Historically, databases didn't "scale" to the web
 - Too small or not enough parallelism
 - E.g., only worked on one machine with multiprocessors, or distributed to "as many as 32 processors"
- One machine not enough - *single point of failure*
 - real fault-tolerant systems can't have this
 - as number of computers go up, probability of one failing goes to 100%
- Because of these distinct, structured relationships between rows and columns in a table, SQL databases are best when you need **ACID compliance**.

Alternatives?

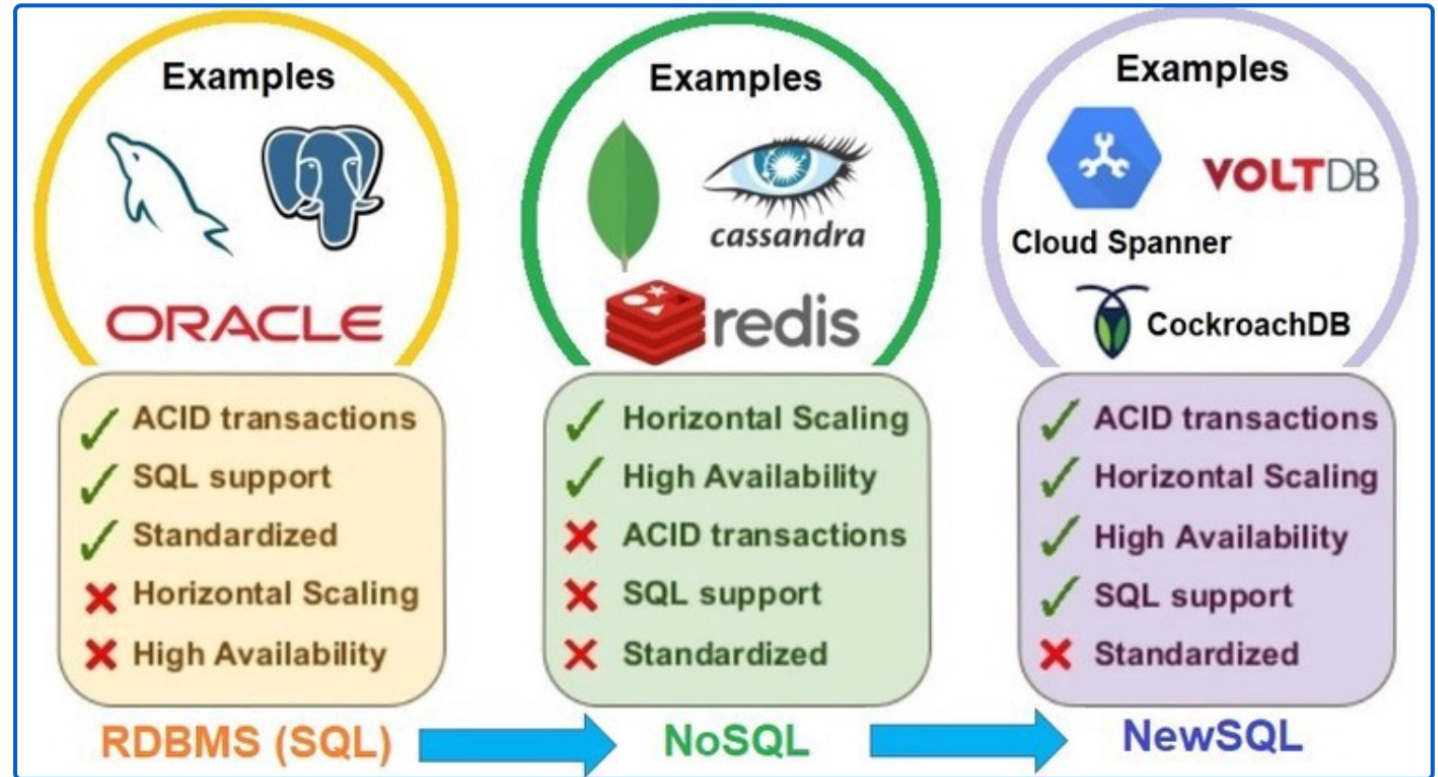
- Result? Companies had to create their own scalable alternatives - mostly "**key value**" stores, gradually evolving to look like SQL databases
 - key-value basically = hash tables, dicts, as in:
dataStore["key"] = "value"
 - Why do these scale?
 - Google:
 - MapReduce
 - see [MapReduce: A major step backwards](#)
 - [MapReduce:Simplified Data Processing on Large Clusters](#)
 - BigTable - [Cloud Bigtable: NoSQL database service](#)
- Amazon: DynamoDB - [Amazon DynamoDB | NoSQL Key-Value Database | Amazon Web Services](#)

Some NoSQL databases went in different directions

- MongoDB - "documents" (JSON)
 - see [Comparing MongoDB vs PostgreSQL](#)
 - also, infamous "MongoDB is web scale":
<https://www.youtube.com/watch?v=b2F-DItXtZs>
 - <http://www.mongodb-is-web-scale.com/>
- Redis - "data structure" server (lists, trees, and so on)
 - <https://redis.io/>
 - *in-memory*, must explicitly persist...
- Others (graphs)
 - [Understanding the Different Types of NoSQL Databases](#)

Future?

- It is only very recently that there are truly scalable SQL databases (Google's Spanner - 2017) - "NewSQL" - [NewSQL](#)
- but some NoSQL databases are just a better *fit* than relational databases



MongoDB Atlas

- MongoDB
- "Atlas"
- [Managed MongoDB Hosting | Database-as-a-Service](https://www.mongodb.com/atlas/database)
 - You can use this free service for your own databases
 - <https://www.mongodb.com/atlas/database>



MongoDB

MongoDB has three basic concepts:

- **Documents**
 - Stores data as “documents” – this is basically JSON
- **Collections**
 - A set of documents
- **Operations**
 - Queries and updates over documents
 - ALL QUERIES ARE IN JSON FORMAT!

```
mongosh mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase

> mongosh "mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase" --apiVersion 1 -
-username timdrichards
Enter password: *****
Current Mongosh Log ID: 625dc66baff3828181dbfa9d
Connecting to:      mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase?appN
ame=mongosh+1.3.1
Using MongoDB:      5.0.7 (API Version 1)
Using Mongosh:      1.3.1

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.insertOne({_id: 1, na
me: "Artemis", age: 19 })
{ acknowledged: true, insertedId: 1 }
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.insertMany([[_id: 2,
name: 'Parzival', age: 17 },
...      { _id: 3, name: 'John', age: 30 },
...      { _id: 4, name: 'Mia', age: 22 }])
{ acknowledged: true, insertedIds: { '0': 2, '1': 3, '2': 4 } }
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> |
```

MongoDB - Operations

Essentially provides CRUD as follows:

- **Create**
 - Insert
- **Read**
 - Find
- **Update**
 - Update: Query + Action
 - “Upsert”: if not found on update, then insert
- **Delete**
 - Delete

```
mongosh mongodb+srv://cluster0.8gq2a.mongodb.net/

> mongosh "mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase"
-username timdrichards
Enter password: *****
Current Mongosh Log ID: 625dc66baff3828181dbfa9d
Connecting to:      mongodb+srv://cluster0.8gq2a.mongodb.net/
ame=mongosh+1.3.1
Using MongoDB:      5.0.7 (API Version 1)
Using Mongosh:      1.3.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.insertOne({
  name: "Artemis", age: 19 })
{ acknowledged: true, insertedId: 1 }
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.find().pretty()
  name: 'Parzival', age: 17 },
...      { _id: 3, name: 'John', age: 30 },
...      { _id: 4, name: 'Mia', age: 22 })
{ acknowledged: true, insertedIds: { '0': 2, '1': 3, '2': 4 } }
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> |
```

MongoDB - Operations

Rich set of commands:

<https://www.mongodb.com/docs/manual/crud>

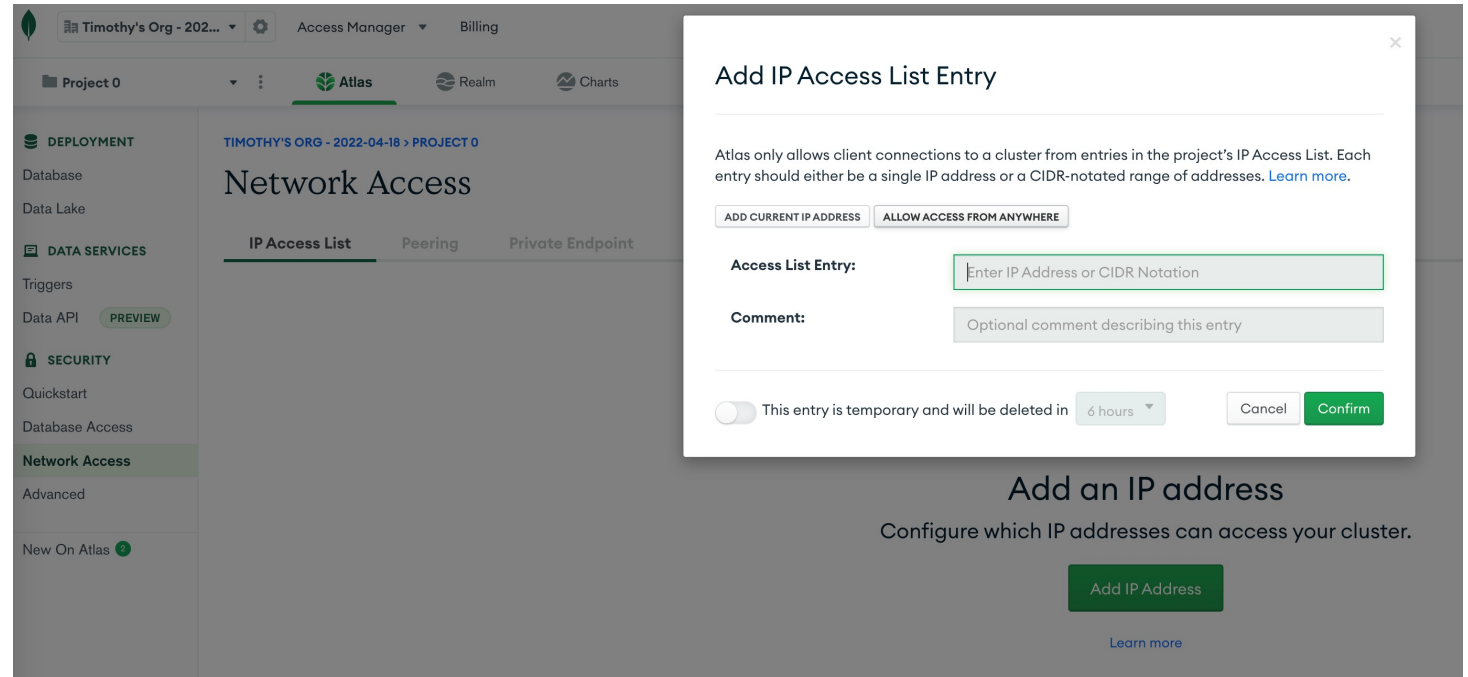
```
db.users.insertOne(
  {
    name: "sue",
    age: 26,
    status: "pending"
  }
)
```

← collection

← field: value
← field: value
← field: value } document

Atlas – Register & Login

- <https://www.mongodb.com/atlas/database>
- Need to register an account and select free tier...
- Also, need to select “Network” and “allow access from anywhere”



Database Deployments

- Select database deployments
- Click on “Connect”

Project 0

Atlas

Realm

Charts

TIMOTHY'S ORG - 2022-04-18 > PROJECT 0

Database Deployments

Find a database deployment...

Cluster0 [Connect](#) [View Monitoring](#) [Browse Collections](#) [...](#)

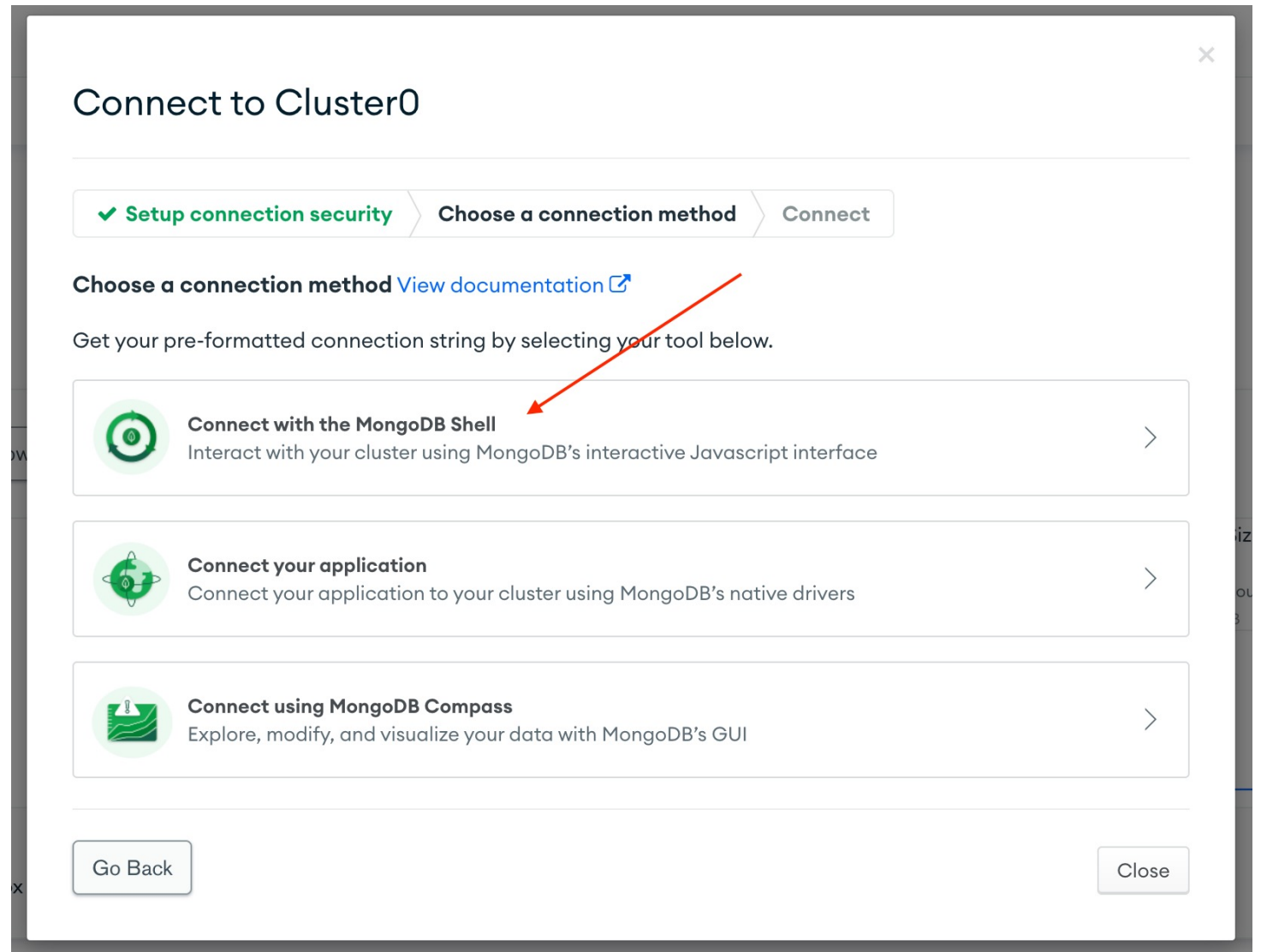
R 0 **W 0** **Connections 6.0**

Last 2 hours 0.007/s Last 2 hours 9.0

VERSION	REGION	CLUSTER TIER	TYPE
5.0.7	AWS / N. Virginia (us-east-1)	M0 Sandbox (General)	Replica Set - 3 nodes

mongosh

- Select database deployments
- Click on “Connect”
- Connect with MongoDB Shell (you need to download **mongosh**)



mongosh

- Select database deployments
- Click on “Connect”
- Connect with MongoDB Shell (you need to download **mongosh**)
- Grab connection command string
- This will be unique to you

×

Connect to Cluster0

✓ Setup connection security

✓ Choose a connection method

Connect

I do not have the MongoDB Shell installed

I have the MongoDB Shell installed

1

Select your operating system and download the mongosh

🍏 macOS

▼

Install via Homebrew

brew install mongosh

📄

Homebrew is a package manager for macOS. [Install Homebrew](#)

2

Run your connection string in your command line

Use this connection string **in your application**:

mongosh "mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase" --apiVersion 1
--username timdrichards

📄

Replace **myFirstDatabase** with the name of the database that connections will use by default. You will be prompted for the password for the Database User, **timdrichards**. When entering your password, make sure all special characters are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

Go Back

Close

mongosh

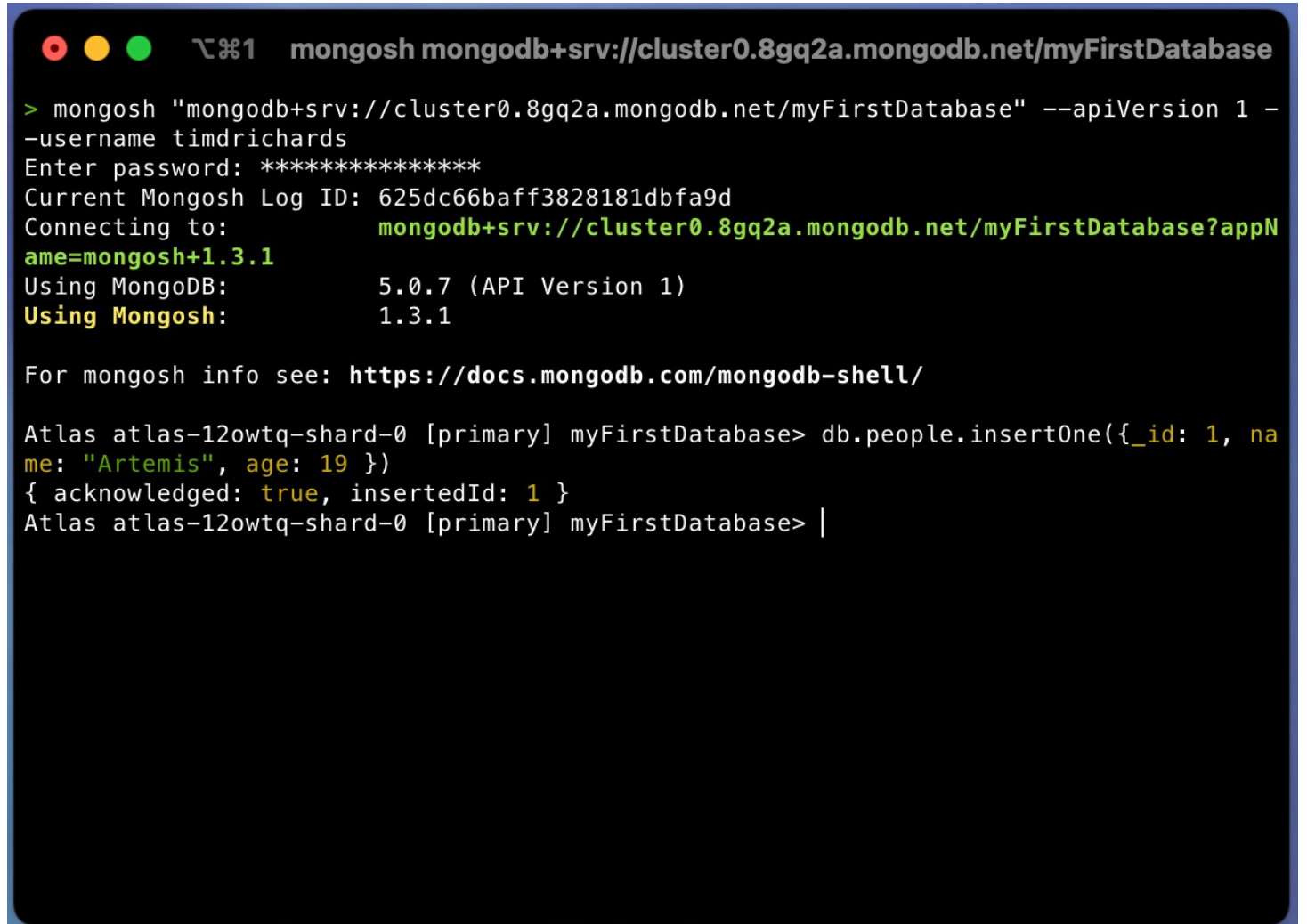
- Select database deployments
- Click on “Connect”
- Connect with MongoDB Shell (you need to download **mongosh**)
- Grab connection command string
- This will be unique to you

```
❯ mongosh mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDa...  
  
> mongosh "mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase" --apiVersion 1 --username timdrichards  
Enter password: *****  
Current Mongosh Log ID: 625dbed0b4b5cb55e7a503f8  
Connecting to:      mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase?appName=mongosh+1.3.1  
Using MongoDB:      5.0.7 (API Version 1)  
Using Mongosh:      1.3.1  
  
For mongosh info see: https://docs.mongodb.com/mongodb-shell/  
  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> |
```

Basic Operations

- `db.collection.insertOne()`

```
db.people.insertOne({
  _id: 1,
  name: 'Artemis',
  age: 19
})
```



```
mongosh mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase
> mongosh "mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase" --apiVersion 1 -
-username timdrichards
Enter password: *****
Current Mongosh Log ID: 625dc66baff3828181dbfa9d
Connecting to:      mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase?appN
ame=mongosh+1.3.1
Using MongoDB:      5.0.7 (API Version 1)
Using Mongosh:      1.3.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.insertOne({_id: 1, na
me: "Artemis", age: 19 })
{ acknowledged: true, insertedId: 1 }
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> |
```

Basic Operations

- `db.collection.insertMany()`

```
db.people.insertMany({
  _id: 2,
  name: 'Parzival',
  age: 17
},
{
  _id: 3,
  name: 'John',
  age: 30
}, ...)
```

```
mongosh mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase

> mongosh "mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase" --apiVersion 1 -
-username timdrichards
Enter password: *****
Current Mongosh Log ID: 625dc66baff3828181dbfa9d
Connecting to:      mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase?appName=mongosh+1.3.1
Using MongoDB:      5.0.7 (API Version 1)
Using Mongosh:      1.3.1

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.insertOne({_id: 1, name: "Artemis", age: 19 })
{ acknowledged: true, insertedId: 1 }
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.insertMany([ { _id: 2, name: 'Parzival', age: 17 },
...      { _id: 3, name: 'John', age: 30 },
...      { _id: 4, name: 'Mia', age: 22 } ])
{ acknowledged: true, insertedIds: { '0': 2, '1': 3, '2': 4 } }
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> |
```

Basic Operations

- `db.collection.find()`

```
db.people.find({ _id: 4 })
```

```
db.people.find({  
  age: { $gt: 21 }  
})
```

```
mongosh mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase  
  
> mongosh "mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase" --apiVersion 1 -  
-username timdrichards  
Enter password: *****  
Current Mongosh Log ID: 625dc66baff3828181dbfa9d  
Connecting to:      mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase?appName=mongosh+1.3.1  
Using MongoDB:      5.0.7 (API Version 1)  
Using Mongosh:      1.3.1  
  
For mongosh info see: https://docs.mongodb.com/mongodb-shell/  
  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.insertOne({_id: 1, name: "Artemis", age: 19 })  
{ acknowledged: true, insertedId: 1 }  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.insertMany([ { _id: 2, name: 'Parzival', age: 17 },  
... { _id: 3, name: 'John', age: 30 },  
... { _id: 4, name: 'Mia', age: 22 } ] )  
{ acknowledged: true, insertedIds: { '0': 2, '1': 3, '2': 4 } }  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.find({_id: 4})  
[ { _id: 4, name: 'Mia', age: 22 } ]  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> |
```

Basic Operations

- `db.collection.updateOne()`

```
db.people.updateOne(  
  {  
    name: 'Mia'  
  },  
  { $set: { age: 23 } })
```

```
mongosh mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase  
  
me: "Artemis", age: 19 })  
{ acknowledged: true, insertedId: 1 }  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.insertMany([ { _id: 2,  
  name: 'Parzival', age: 17 },  
  ... { _id: 3, name: 'John', age: 30 },  
  ... { _id: 4, name: 'Mia', age: 22 } ] )  
{ acknowledged: true, insertedIds: { '0': 2, '1': 3, '2': 4 } }  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.find({ _id: 4 })  
[ { _id: 4, name: 'Mia', age: 22 } ]  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.find({ age: { $gt: 21  
  } })  
[ { _id: 3, name: 'John', age: 30 }, { _id: 4, name: 'Mia', age: 22 } ]  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.update  
db.people.updateMany db.people.updateOne  
  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.updateOne({ name: 'Mia  
' }, { $set: { age: 23 } })  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.find({ age: { $gt: 21  
  } })  
[ { _id: 3, name: 'John', age: 30 }, { _id: 4, name: 'Mia', age: 23 } ]  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> |
```

Basic Operations

- `db.collection.deleteOne()`

```
db.people.deleteOne(  
  {  
    name: 'John'  
  })
```

```
mongosh mongodb+srv://cluster0.8gq2a.mongodb.net/myFirstDatabase  
  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.update  
db.people.updateMany db.people.updateOne  
  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.updateOne({name: 'Mia'  
'}, { $set: { age: 23 } })  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.find({ age: { $gt: 21  
}})  
[ { _id: 3, name: 'John', age: 30 }, { _id: 4, name: 'Mia', age: 23 } ]  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.deleteOne({ name: 'Jo  
hn' })  
{ acknowledged: true, deletedCount: 1 }  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.find({ age: { $gt: 21  
}})  
[ { _id: 4, name: 'Mia', age: 23 } ]  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> db.people.find({})  
[  
  { _id: 1, name: 'Artemis', age: 19 },  
  { _id: 2, name: 'Parzival', age: 17 },  
  { _id: 4, name: 'Mia', age: 23 }  
]  
Atlas atlas-12owtq-shard-0 [primary] myFirstDatabase> |
```

Code Examples

- 01-crud-node-nosql
- 02-crud-express-nosql
- 03-deploy-heroku