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

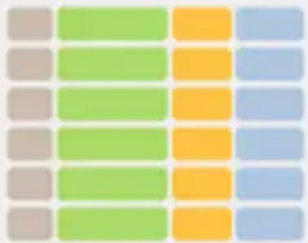
Broadly speaking, there are two kinds of databases: SQL and "NoSQL". NoSQL databases are diverse.

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.

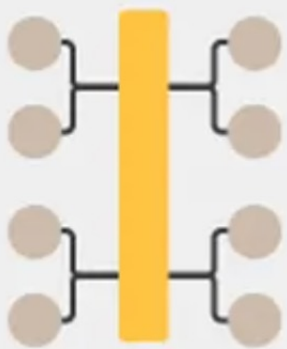
Others, like MongoDB, are "document oriented", and directly let you manage JSON - we'll talk about MongoDB today.

SQL Database

Relational

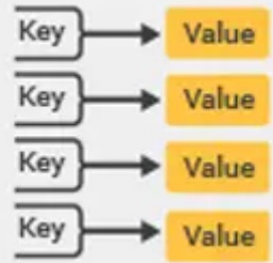


Analiticals (OLAP)



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?

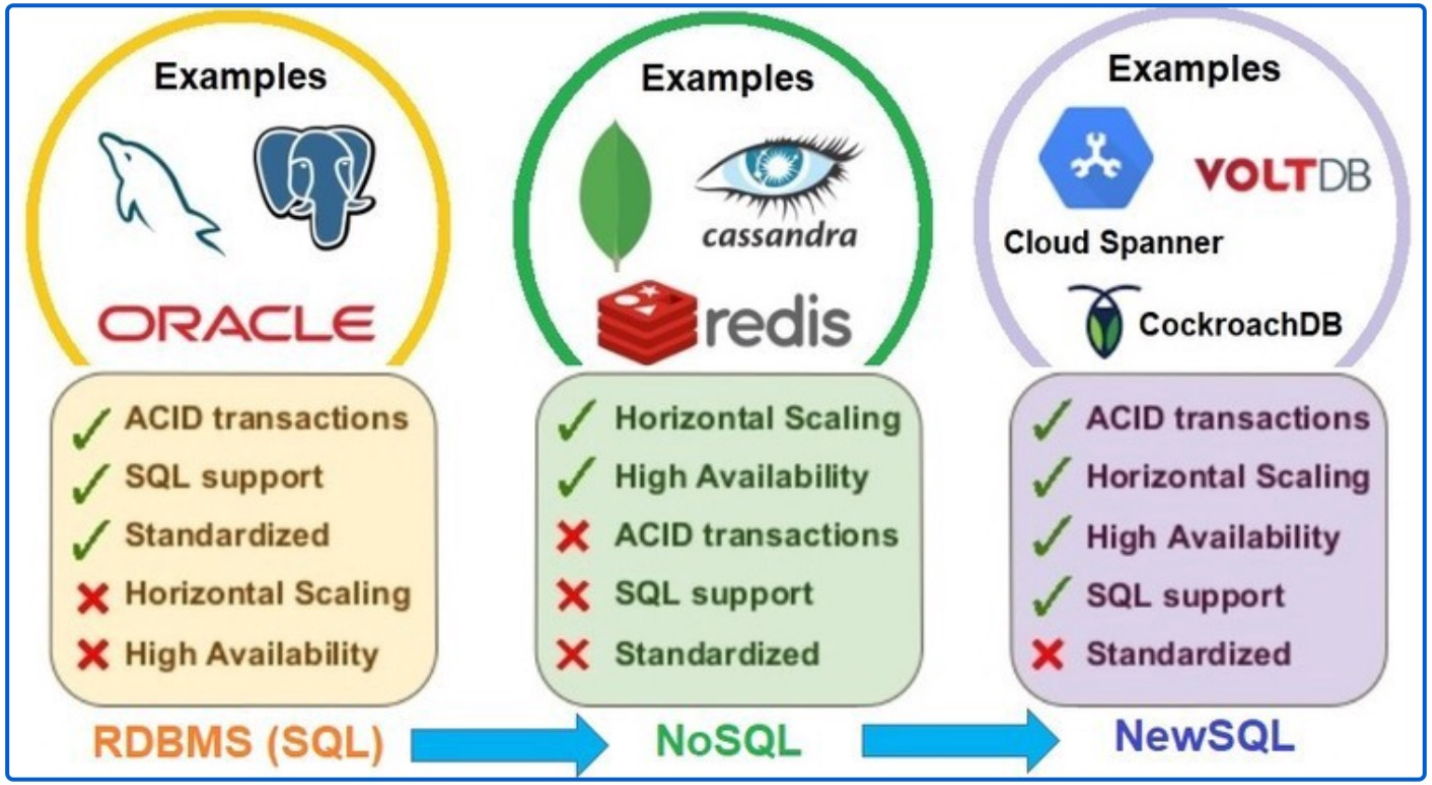
- Result? Companies had to create their own scalable alternatives - mostly "***key value***" stores, gradually evolving to look like SQL databases
  - key-value basically = hash tables, dicts, as in:
    dataStore["key"] = "value"
  - Why do these scale?
  - Google:
    - MapReduce
      - see MapReduce: 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



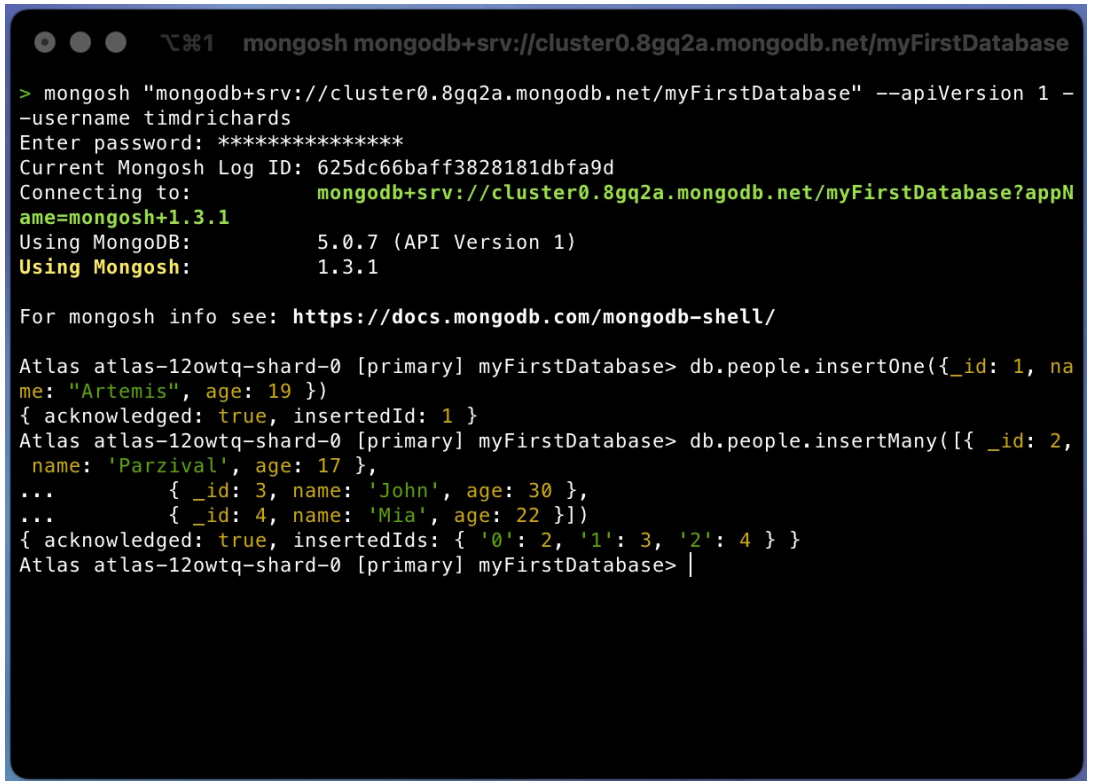
MongoDB Atlas

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

MongoDB

MongoDB has three basic concepts:

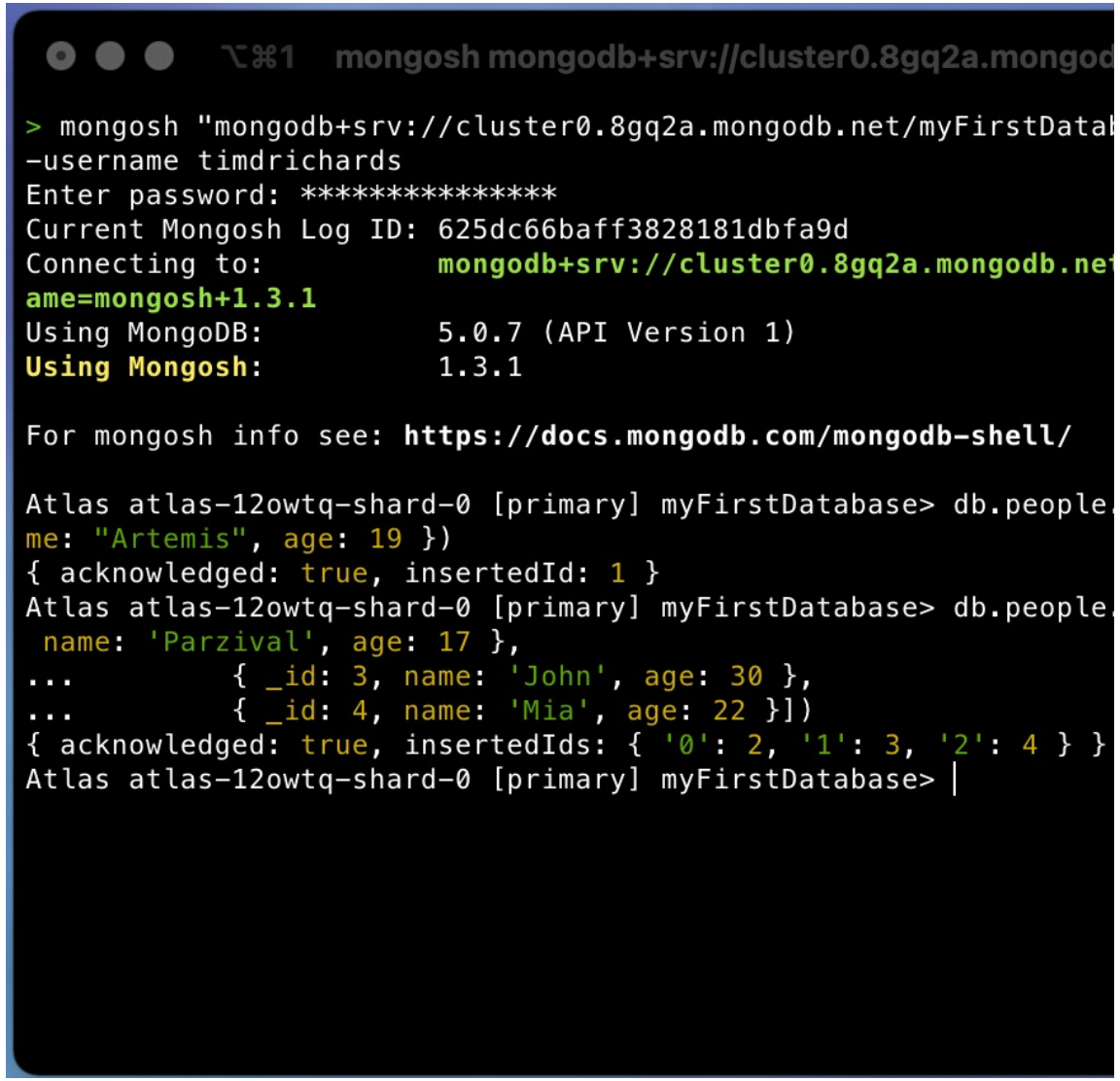
- **Documents**
  - Stores data as “documents” – this is basically JSON
- **Collections**
  - A set of documents
- **Operations**
  - Queries and updates over documents
  - ALL QUERIES ARE IN JSON FORMAT!



MongoDB - Operations

Essentially provides CRUD as follows:

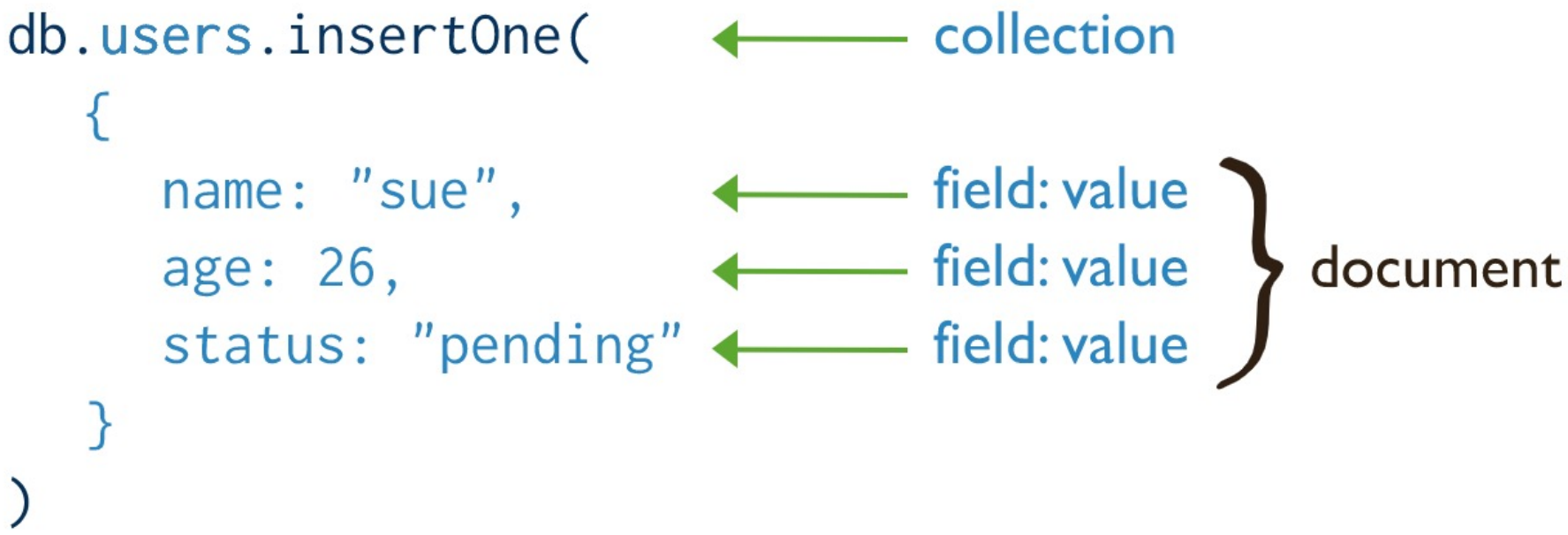
- **Create**
  - Insert
- **Read**
  - Find
- **Update**
  - Update: Query + Action
  - “Upsert”: if not found on update, then insert
- **Delete**
  - Delete



MongoDB - Operations

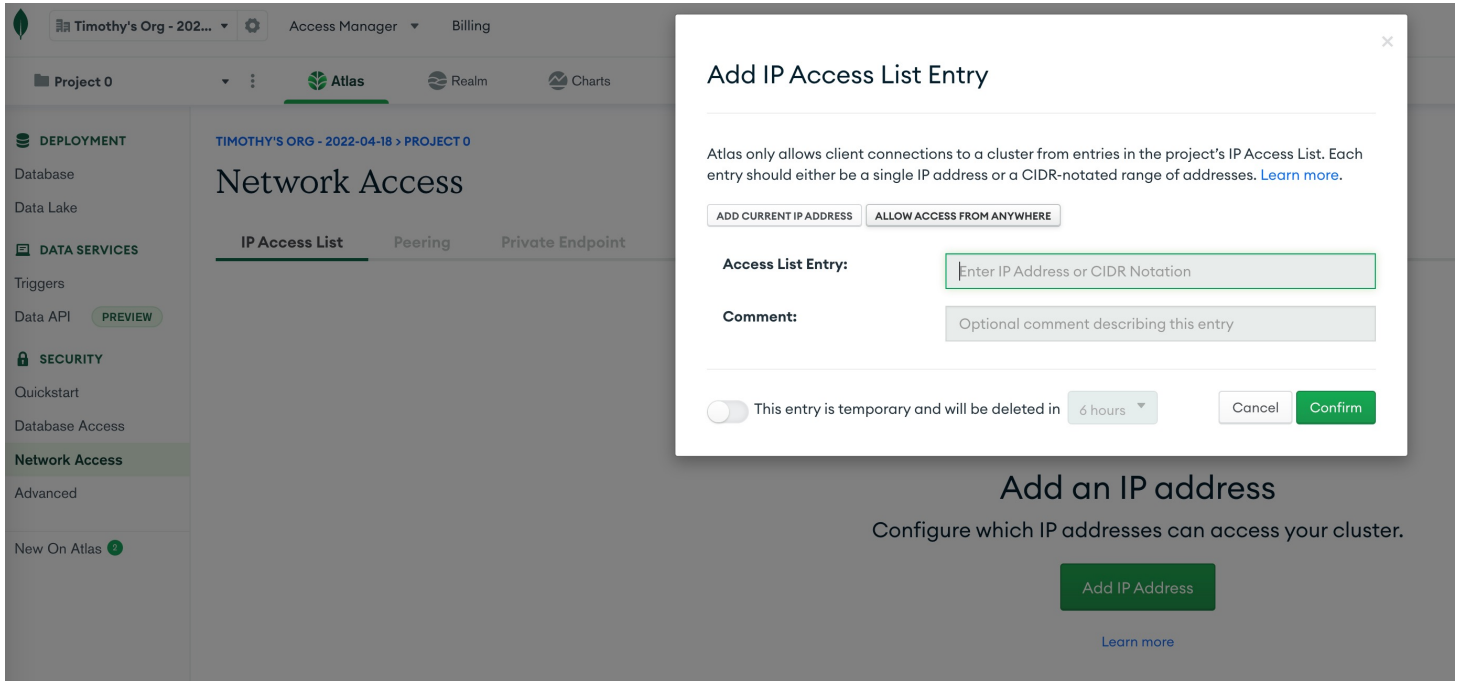
Rich set of commands:

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



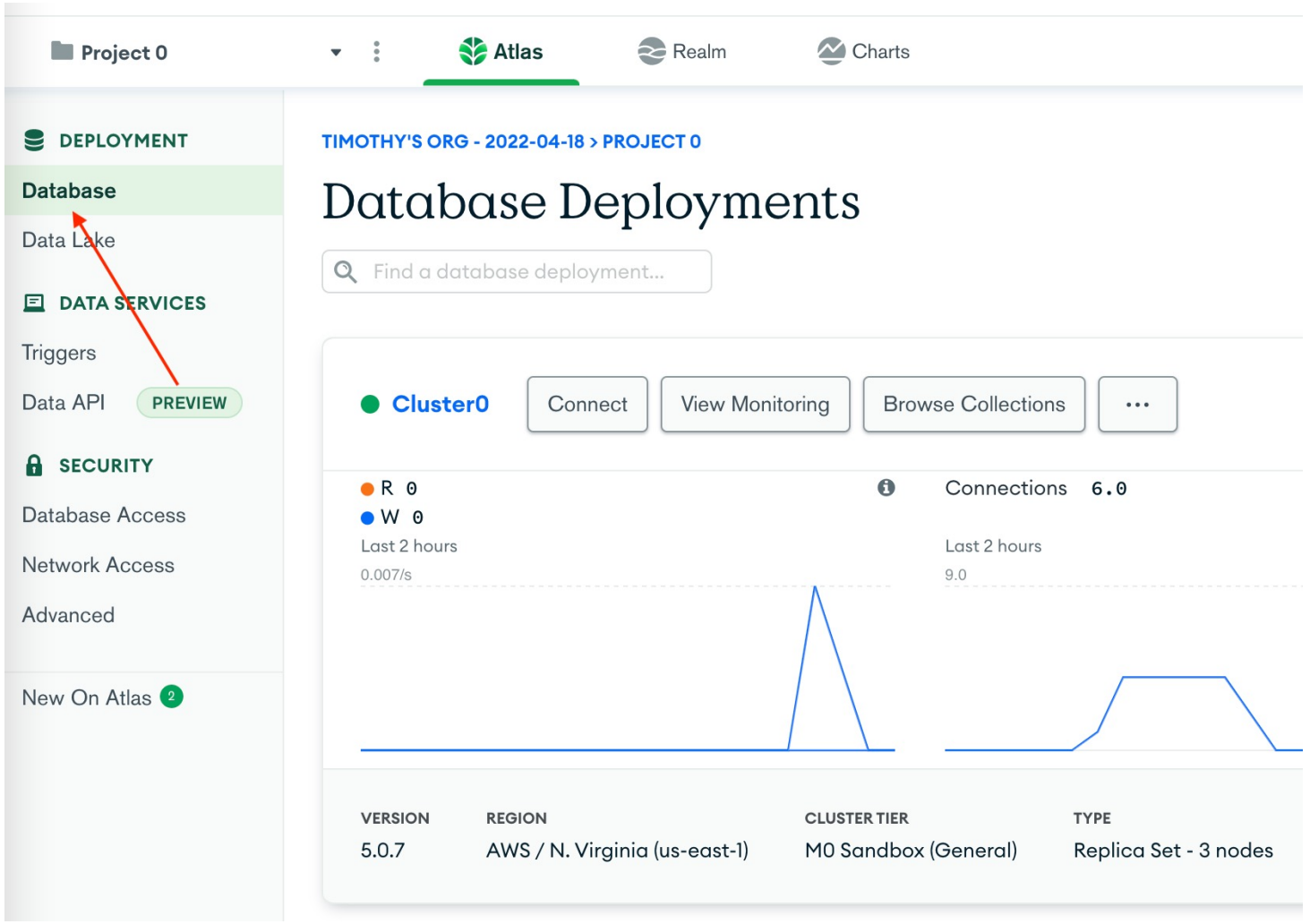
Atlas – Register & Login

- 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”



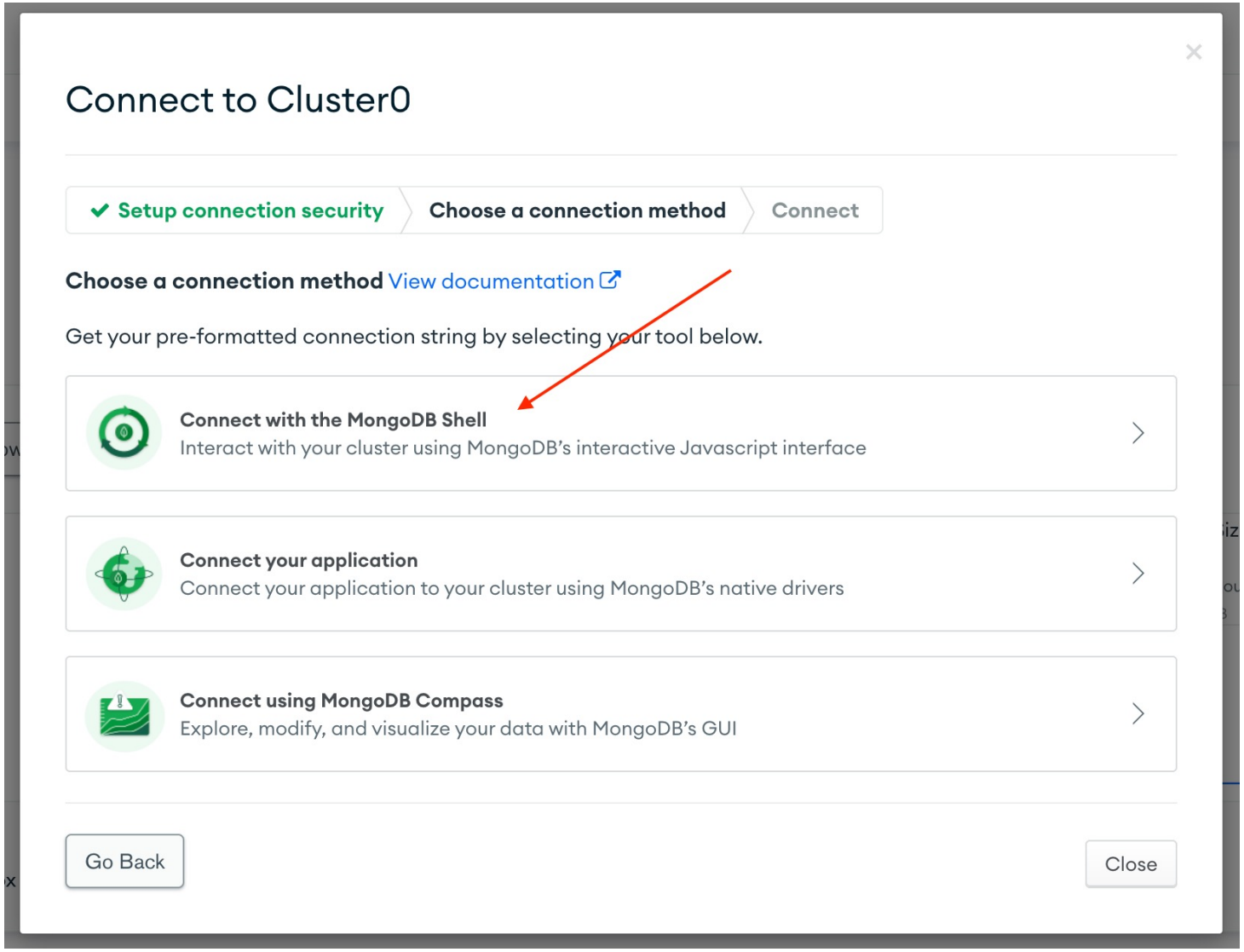
Database Deployments

- Select database deployments
- Click on “Connect”



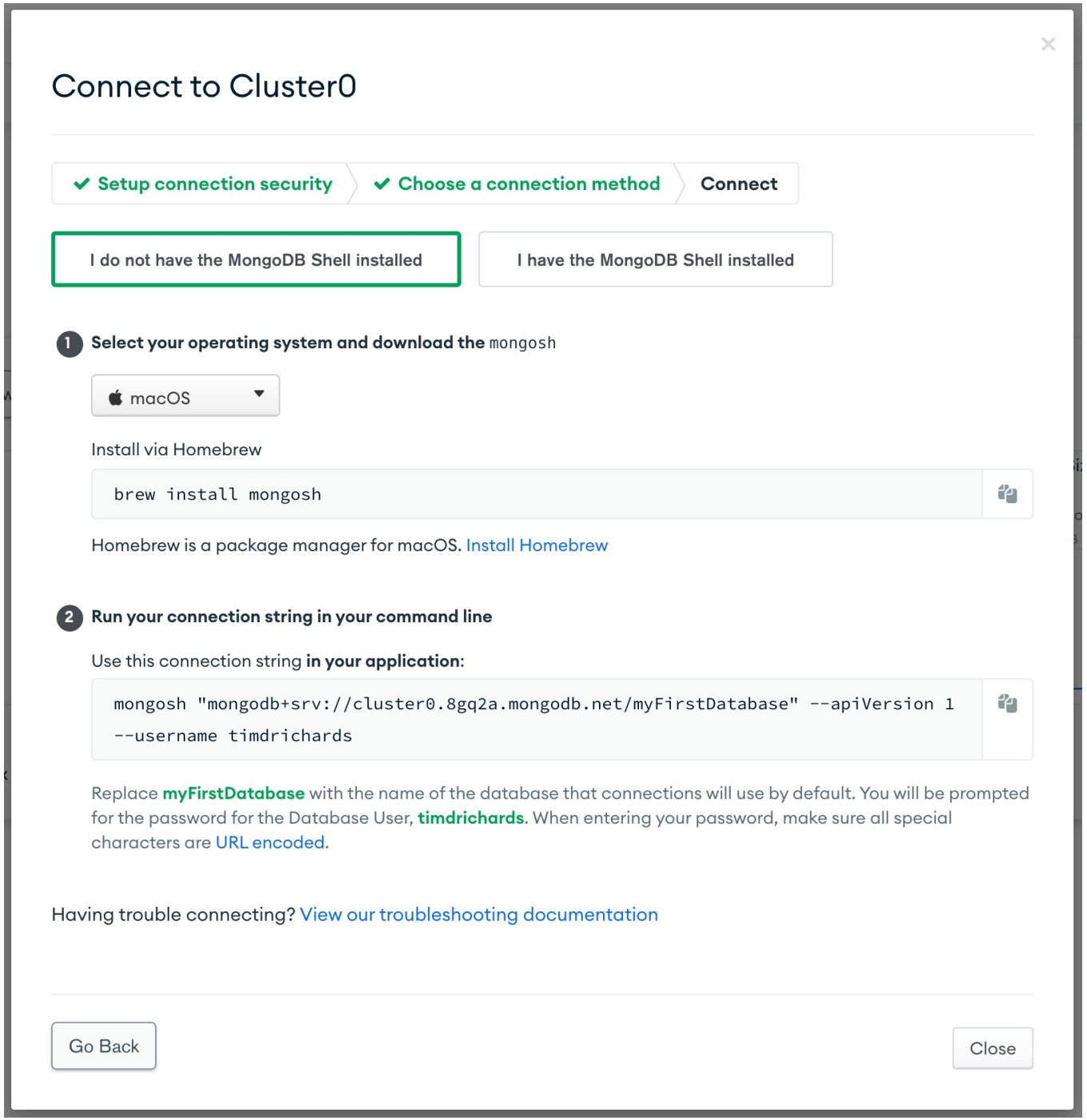
mongosh

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



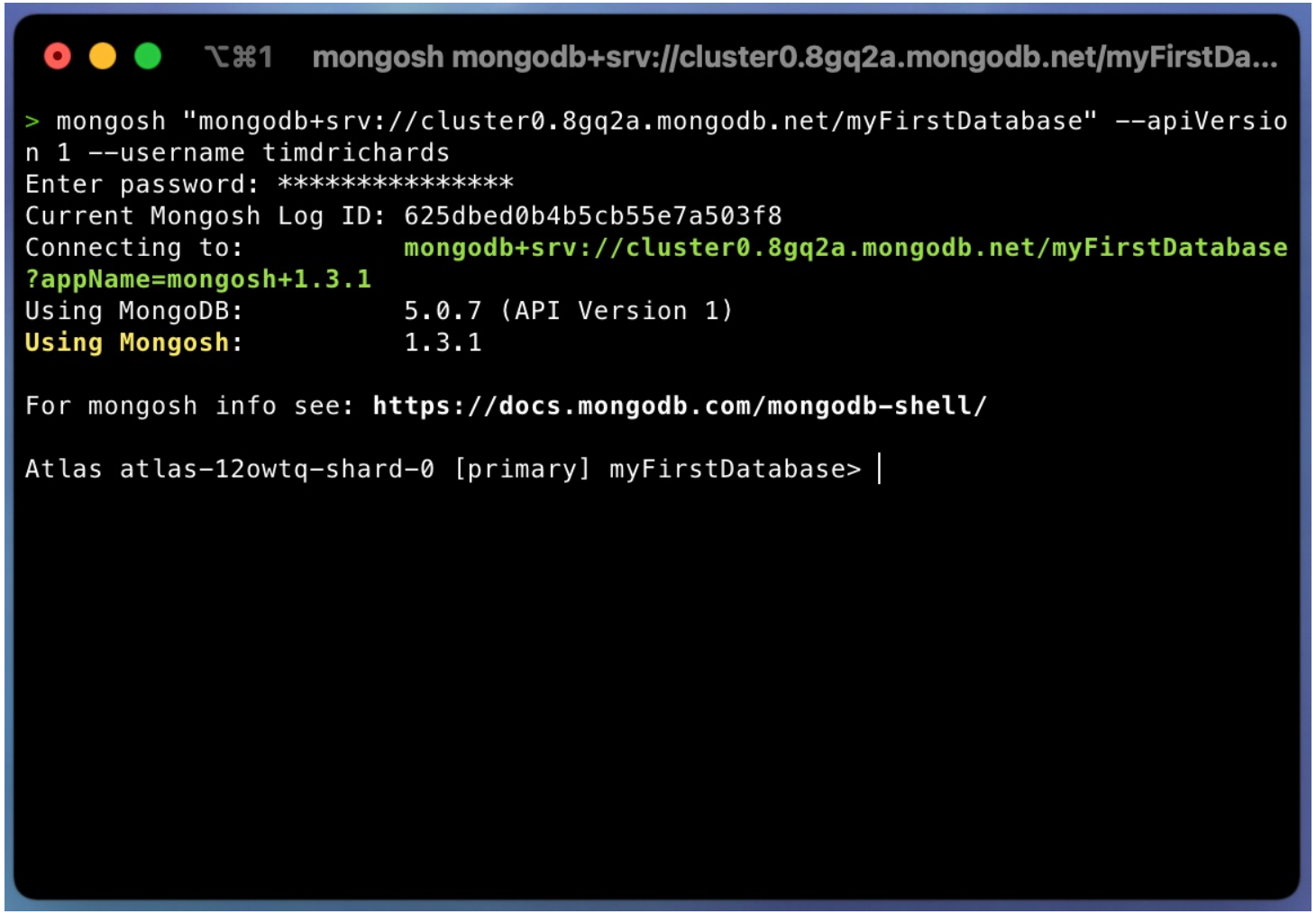
mongosh

- 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



mongosh

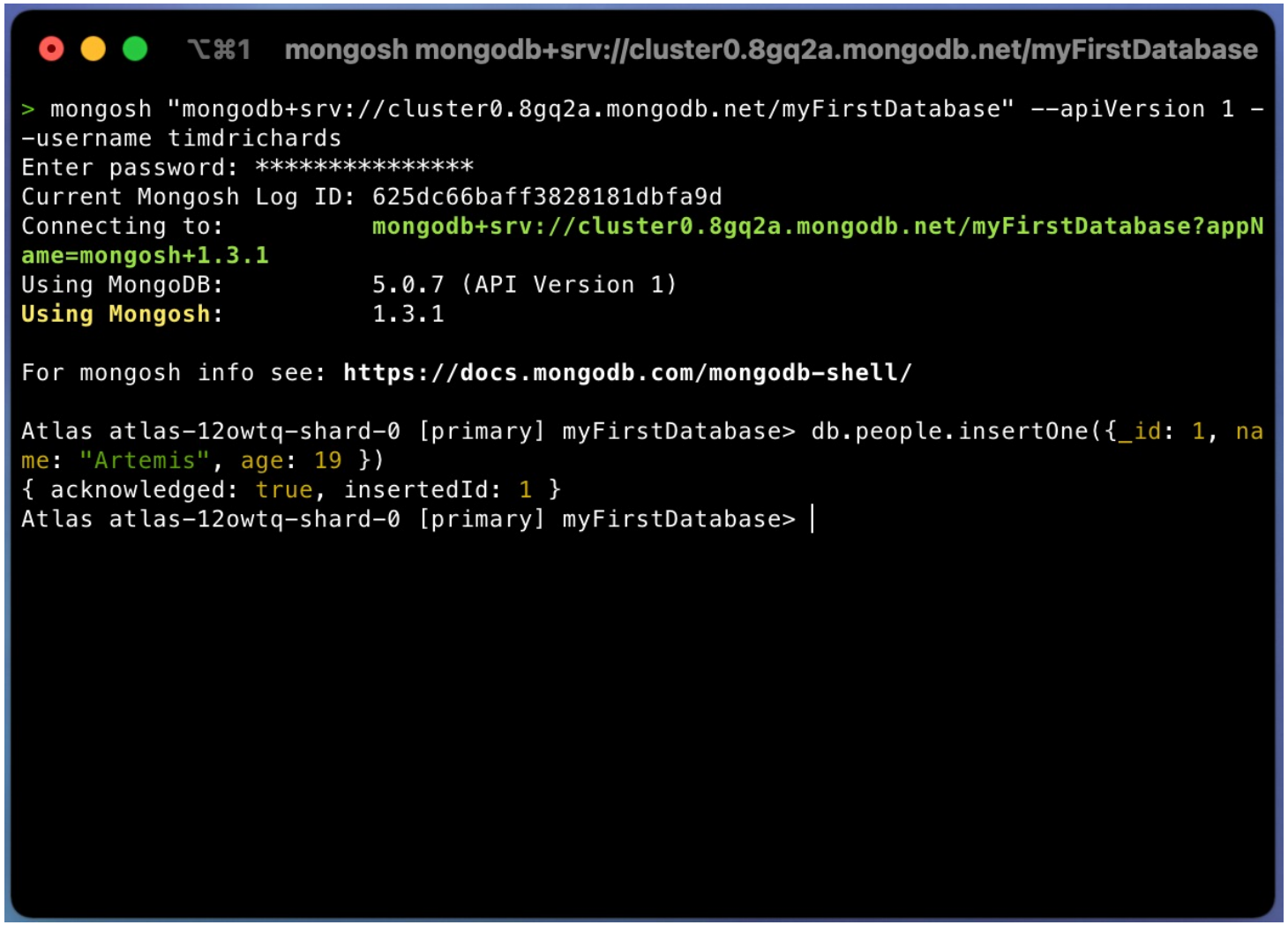
- Select database deployments
- Click on “Connect”
- Connect with MongoDB Shell (you need to download **mongosh**)
- Grab connection command string
- This will be unique to you



Basic Operations

- db.collection.insertOne()

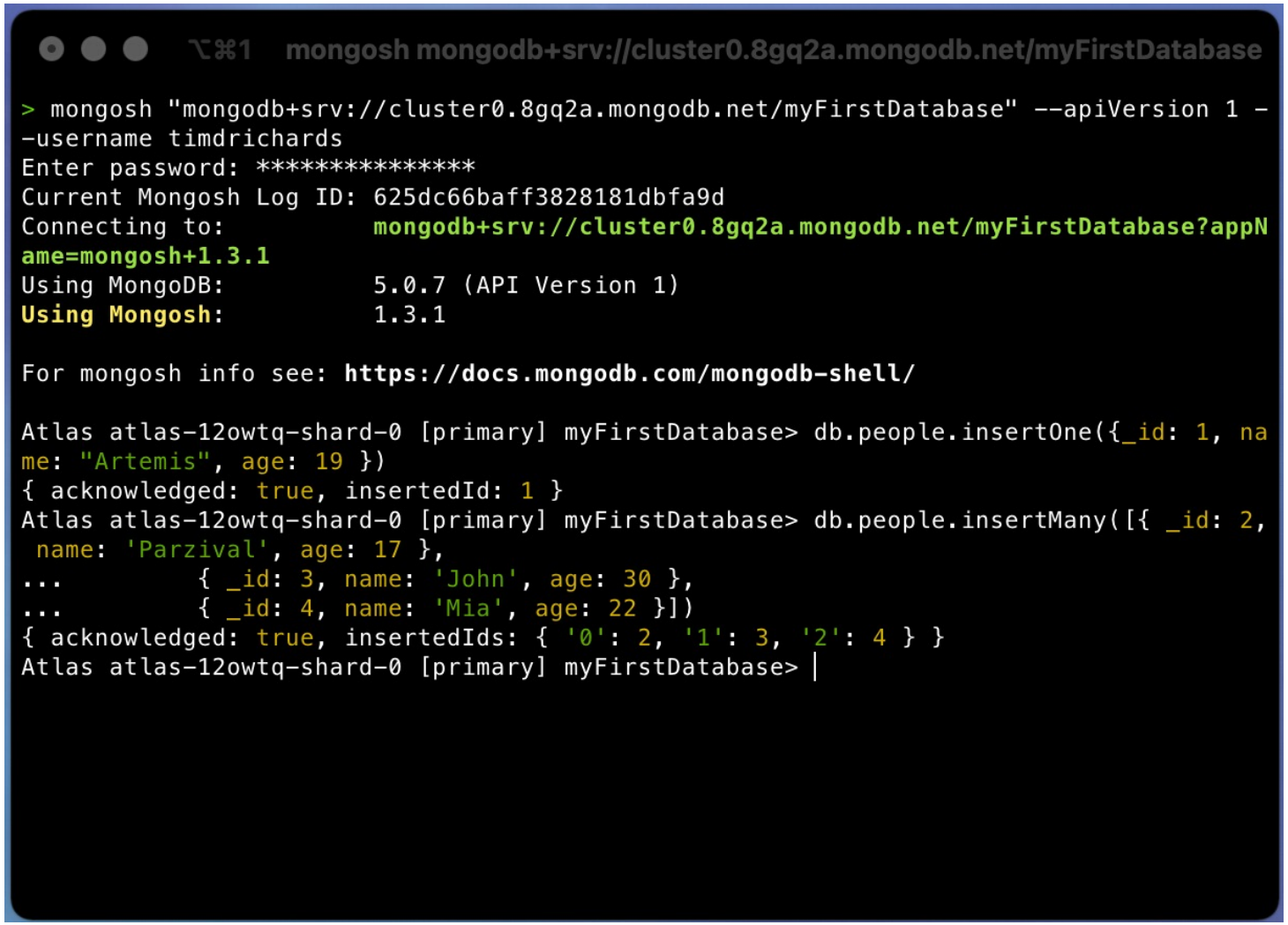
```
db.people.insertOne({
  _id: 1,
  name: ‘Artemis’,
  age: 19
})
```



Basic Operations

- db.collection.insertMany()

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

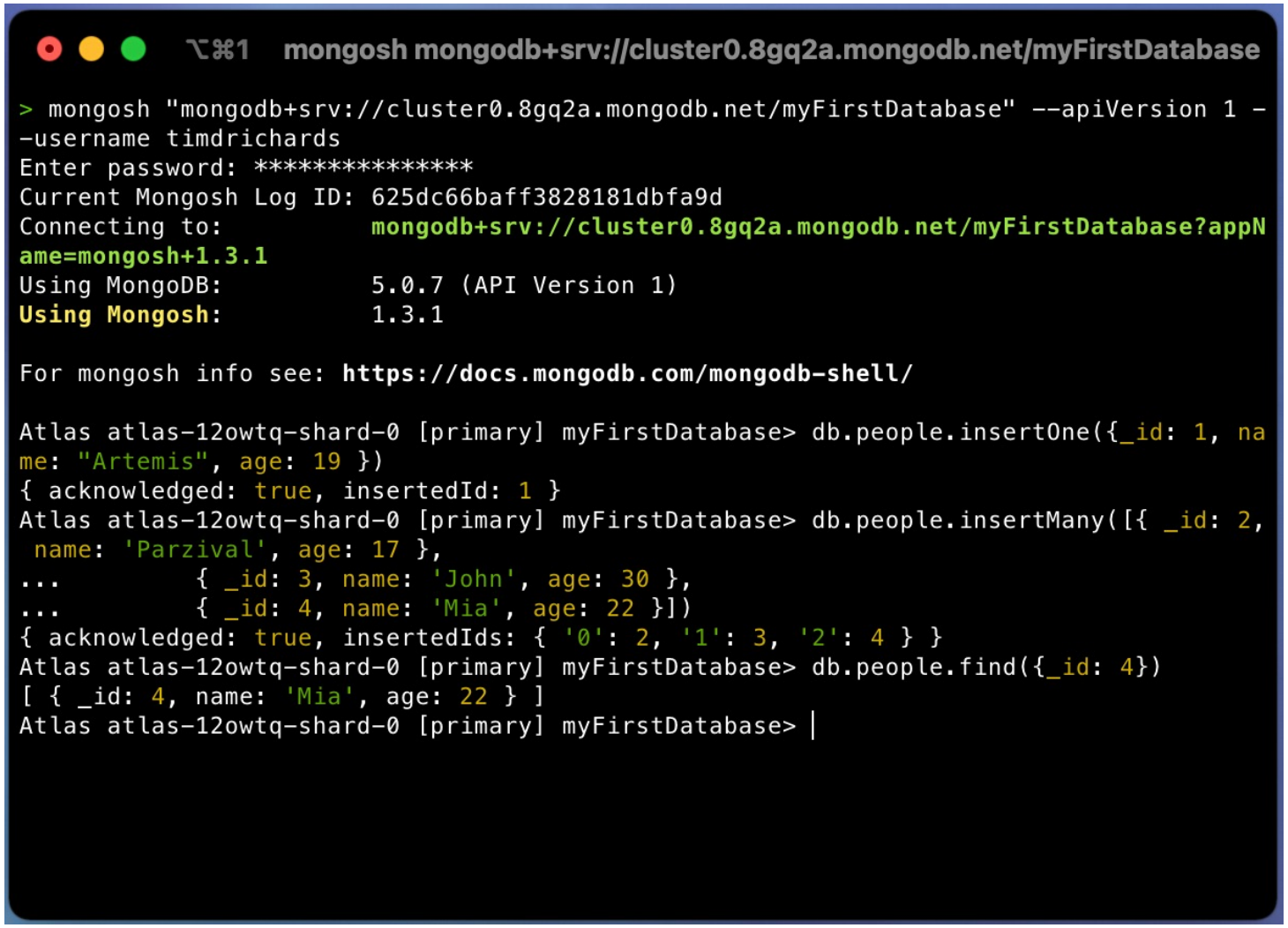


Basic Operations

- db.collection.find()

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

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



Basic Operations

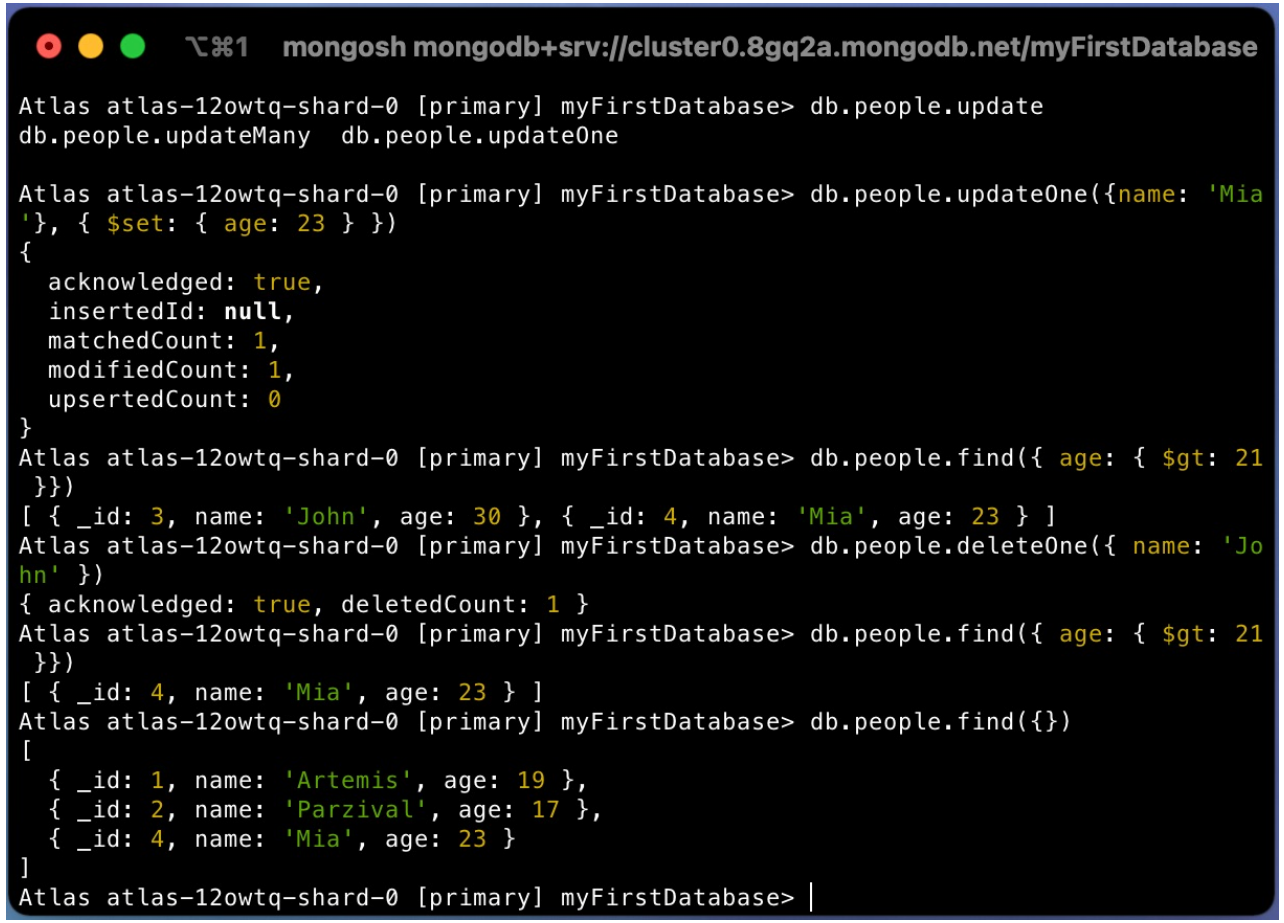
- db.collection.updateOne()

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

Basic Operations

- db.collection.deleteOne()

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



Code Examples

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