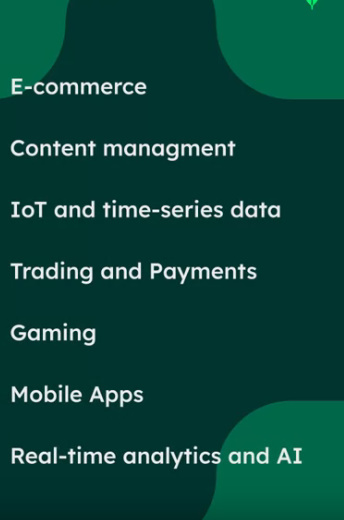
* Atlas este deja un DbaaS, si el este cel care se ocupa deja de configurarea la Mongo DB
* In Atlas, DB deja are Replication setat, numit **Replica Set,**ceea ce inseamna ca datele sunt stocate pe mai multe servere, un concept fundamental in NoSQL si deci MongoDB
* **Cluster** – un grup de calculatoare, sau noduri, care lucreaza impreuna, in paralel, pentru a atinge un scop comun.
* Un cluster combina puterea la toate nodes(pcs) si asa lucrul e facut mult mai rapid
* Acest tip de cluster se numeste **Shared Cluster**, cunoscut si ca **horizontal scaling**

**MongoDB Replica Set**

* Un MongoDB Cluster trebuie sa aiba el mult un node principal sau primar si un set de noduri secundare pentru a fi considerat **replica set**
* Acest primary node primeste toate write operations. Toate modificarile in datele nodului principal sunt inregistrate intr-o colectie speciala, numita **operation log(oplog)**
* Rolul la nodurile secundare este de a copia datele din nodul principal si de a se asigura ca datele pe care nodurile secundare le copie sa fie exact aceleasi ca in nodul principal. Tototdata, ele replicheaza operations log de la primary node
* Daca la un moment dat, vreun secondary node cade, nu e nimic, totul se face ca deobicei, dar fara el
* Daca insa nodul principal cade, unul din nodurile secundare ii va lua locul
* In mod normal, clientii comunica cu nodul primar, dar se poate seta ca sa comunice si cu vreun nod secundar

**MongoDB Structure**

* Fiecare JSON object este un document separat
* O **Collection** contine mai multe documente ce au ceva in comun
* Un document poate avea orice date, nu trebuie ca toate sa aiba aceeasi structura
* **Document** – basic unit of data
* **Database** – container pentru collections
* MongoDB e folosit in:



**Document Model**

* Desi documents sunt aratate in JSON format, ele sunt pastrate in **BSON** format
* **BSON** – Binary JSON
* **BSON** este o extensie la JSON ce ofera functiuni suplimentare in MongoDB si ofera suport pentru tipuri de date nedisponibile in JSON
* Deci, ca JSON, BSON suporta boolean, string, number, array, object etc. dar mai suporta si Dates, ObjectId, numbers si altele
* **ObjectId** – tip de date special in MongoDB pentru a crea un id unic per document
* Ca primary key, fiecare document trebuie sa aiba un field:

“\_id” : ....

Daca el nu e pus de noi, MongoDB il pune automat si creaza un ID

* Deoarece MongoDB ofera documentelor o structura flexibila, putem chiar avea documente cu structura diferita in acceasi collection
* In RDMBS, o adaugarea chiar si a unei coloane suplimentare la tabel e foarte anevoioasa, dar in MongoDB e foarte usoara
* Daca vrem sa avem anumite constrangeri asupra la date si schema, putem adauga Schema Validation
* **Optional Schema Validation** – set de constrangeri asupra structurii documentelor intr-o colectie

**Fields name restrictions**

* Numele \_id este deja rezervat pentru primary key, si valoarea lui trebuie sa fie unica in colectie, este imutabil si poate fi de orice tip, in afara de array
* \_id mereu e primul field!!!
* Fields pot contine . si $
* Putem accesa un item dintr-un field ce e un array asa **field.index**:

|  |
| --- |
| { |
| ... |
| contribs: [ **"Turing machine"**, **"Turing test"**, **"Turingery"** ], |
| ... |
| } |

contribs.2

* Pentru a accesa un field dintr-un subdocument, folosim ,

|  |
| --- |
| { |
| ... |
| name: { first: **"Alan"**, last: **"Turing"** }, |
| contact: { phone: { type: **"cell"**, number: **"111-222-3333"** } }, |
| ... |
| } |

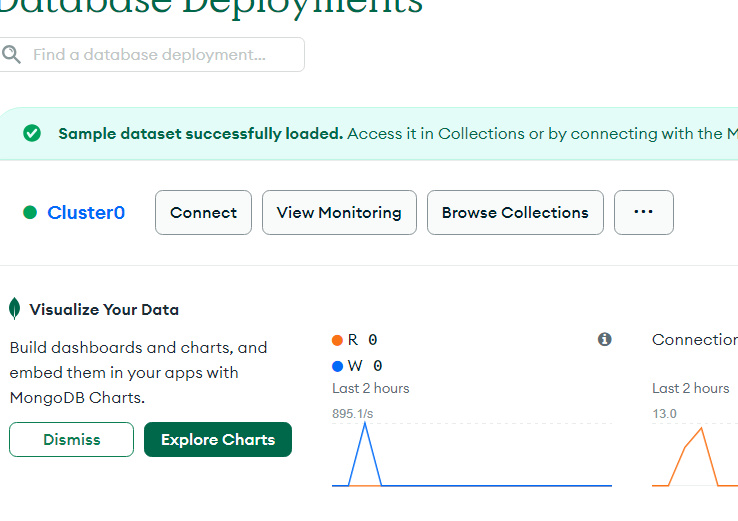
name.first

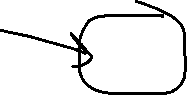
**Document field order**

* Cand comparam documente,e ordinea la fielduri e importanta
* {a: 1, b: 1} is equal to {a: 1, b: 1}
* {a: 1, b: 1} is not equal to {b: 1, a: 1}
* Dar, in timpul la un query, putem rearanja ordinea

**Connection Strings**

* **Connection Strings** – permite sa ne conectam la cluster si sa lucram cu datele. El descrie host ce va fi folosit si optiunile pentru conectare la MongoDB
* Connection Strings poate fi folosit pentru a ne conecta din Compass, Shell sau alte aplicatii
* Apasam pe connect pentru info de conectare





**Connection string example:**

mongosh "mongodb+srv://cluster0.yo0akow.mongodb.net/" --apiVersion 1 --username edikutsu2002

**Shell**

* Atentie!Cand te conectezi cu shell:

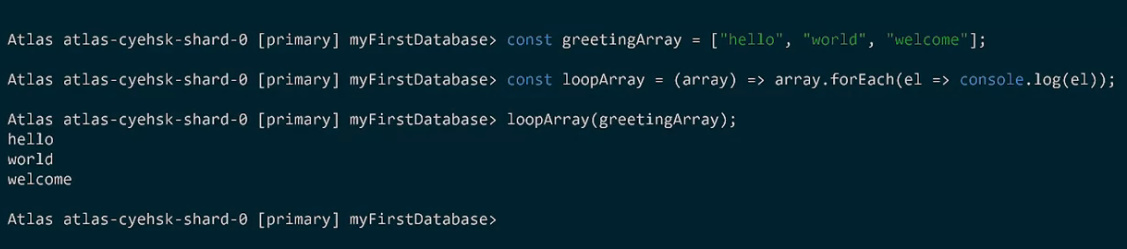
Pune in $Path adresa la bin din shell

Scrie connection string in CMD, nu direct in shell!!!

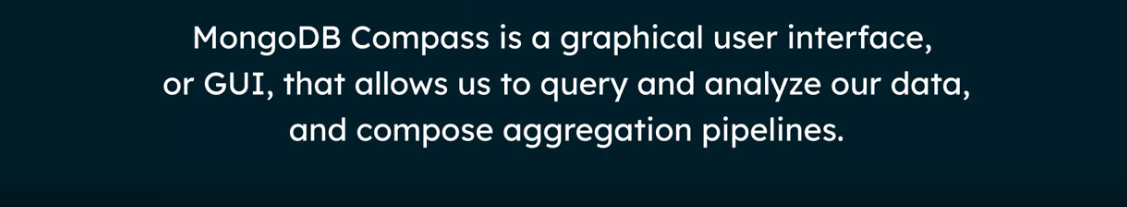
* MongoDB Shell este un Node.js environment , care ofera acces la:



Putem declara in el variabile, functii etc. si apoi le folosi cum vrem:



**Compass**



Cu compas si shell ne conectam la cluster, nu baza de date! Avem doar mai multe baze de date!!!

**Driver**

* Ofera posibilitatea unei aplicatii de a se conecta la cluster
* [Java | MongoDB](https://www.mongodb.com/developer/languages/java/)
* [MongoDB Course Catalog Homepage | MongoDB University](https://learn.mongodb.com/catalog?labels=%5B%22Language%22%5D&values=%5B%22Java%22%5D)