



DATA MANAGEMENT WORKSHOP

Platform Engineering

Platform Engineering

Table of Contents

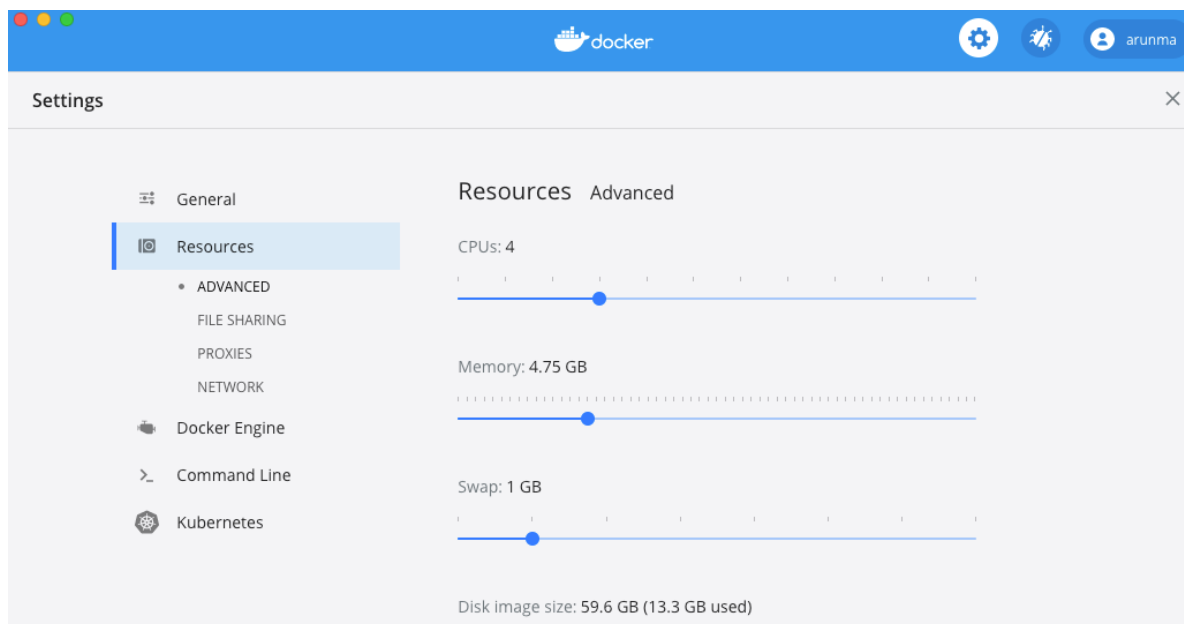
Clone your workshop code from	3
Adjust your docker settings to give at least 4 cores and 3 GB memory.	3
Launch all databases on your local machine using docker-compose	3
ElasticSearch.....	3
Open up Kibana	3
Open ElasticSearch Dev Tools	4
Create index (Table/Collection).....	4
Read all documents	4
Create document with custom id.....	5
Get document by id.....	6
Update documents.....	6
Bulk upload.....	7
Search all documents	9
Search by query.....	10
Delete a Document.....	11
Redis.....	12
Lists.....	12
Push to a List	12
Push multiple values to a List	12
Get values from a List	13
Removing a value from a List	13
Removing a specific value from a List	13
Find the length of the List	13
Sets	13
Push to a Set.....	13
Push multiple values to a Set	13
Listing members in a Set	14
Removing a value from a Set.....	14
Checking membership in a Set	14
Hashes	14
Push to a Hash.....	14
Pushing multiple values to a Hash.....	15
Get value for a key from a Hash	15
Get all values from a Hash.....	15
Removing a value from a Hash.....	16
Neo4J.....	17
Open up Neo4J browser.....	17
Create new Nodes	17
Create new Relationship	17
Query a single node.....	17
Get all nodes and relationships	18
Deleting a node	19
Delete Relationship	19
Delete Nodes.....	19
Load Graph from CSV	20
MongoDB	22
Login to your mongodb docker container using the following command	22
Create/Switch to a database	23
Create a new collection	23
Insert a new document	23

Bulk insert documents.....	23
Query all documents in a collection	25
Find document in a collection matching a criteria	25
Find document in a collection matching multiple criteria.....	25
Removing a document	26
Updating a document.....	26

Clone your workshop code from

<https://github.com/arunma/datamanagement>

Adjust your docker settings to give at least 4 cores and 3 GB memory.



Launch all databases on your local machine using docker-compose

```
cd <location of datamanagement>  
# eg. cd /Users/arunma/projects/data-management
```

```
docker-compose -f docker-compose.yml up --build
```

ElasticSearch

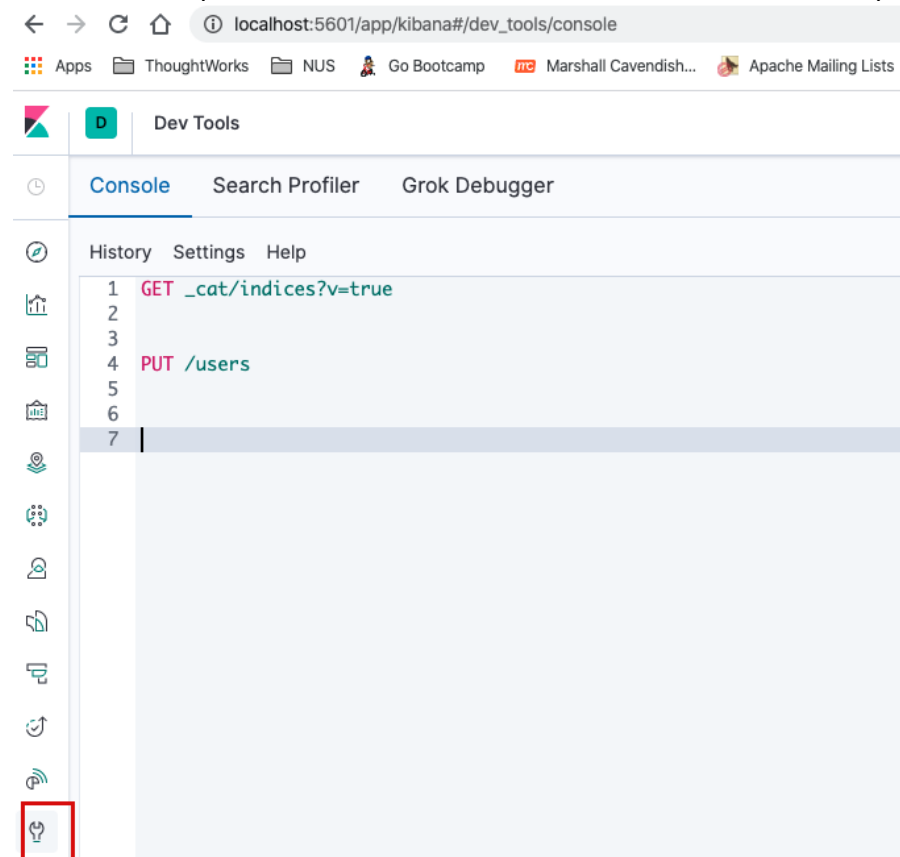
Open up Kibana

on your browser by pointing to

```
http://localhost:5601
```

Open ElasticSearch Dev Tools

Click on the “Spanner” icon to launch the Dev Tools ElasticSearch query page



Create index (Table/Collection)

```
PUT /users
```

You should get back

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "users"
}
```

Read all documents

```
GET /users/_search
```

You should get 0 hits as below:

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 0,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  }
}
```

Create document with custom id

(You can also copy the json from data-management/elastic/users_single.json)

```
PUT /users/_doc/1
{
  "empld": "5e5a766afa73dddb81a1c939",
  "age": 21,
  "eyeColor": "brown",
  "name": "Allison Strong",
  "gender": "male",
  "company": "ENERFORCE",
  "email": "allisonstrong@enerforce.com",
  "phone": "+1 (963) 503-2603",
  "address": {
    "addId": "5e5a766a0e58c8b905902b40",
    "door": 822,
    "street": "Bushwick Place",
    "city": "Castleton",
    "state": "Guam",
    "pin": 2457
  },
  "favoriteFruit": "banana"
}
```

You should get back

```
{
  "_index" : "users",
  "_type" : "_doc",
  "_id" : "1",
  "_version" : 4,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "_seq_no" : 3,
  "_primary_term" : 1
}
```

Get document by id

GET /users/_doc/1

```
{
  "_index" : "users",
  "_type" : "_doc",
  "_id" : "1",
  "_version" : 4,
  "_seq_no" : 3,
  "_primary_term" : 1,
  "found" : true,
  "_source" : {
    "empId" : "5e5a766afa73dddb81a1c939",
    "age" : 21,
    "eyeColor" : "brown",
    "name" : "Allison Strong",
    "gender" : "male",
    "company" : "ENERFORCE",
    "email" : "allisonstrong@enerforce.com",
    "phone" : "+1 (963) 503-2603",
    "address" : {
      "addId" : "5e5a766a0e58c8b905902b40",
      "door" : 822,
      "street" : "Bushwick Place",
      "city" : "Castleton",
      "state" : "Guam",
      "pin" : 2457
    },
    "favoriteFruit" : "banana"
  }
}
```

Update documents

- There's no update in elasticsearch
- The old document is replaced (deleted) with the new one.

Try changing some values for doc 1 and re-PUT the document

PUT /users/_doc/1
{

```
"empId": "5e5a766afa73dddb81a1c939",
"age": 21,
"eyeColor": "brown",
"name": "Allison Strong",
"gender": "male",
"company": "ENERFORCE",
"email": "allisonstrong@enerforce.com",
"phone": "+1 (963) 503-2603",
"address": {
  "addId": "5e5a766a0e58c8b905902b40",
  "door": 822,
  "street": "Bushwick Place",
  "city": "Castleton",
  "state": "Guam",
  "pin": 2457
},
"favoriteFruit": "kiwifruit"
}
```

You must see “updated” in the response

```
{
  "_index" : "users",
  "_type" : "_doc",
  "_id" : "1",
  "_version" : 5,
  "result" : "updated",
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "_seq_no" : 4,
  "_primary_term" : 1
}
```

Bulk upload

You can also find the json used in this exercise at [data-management/elastic/users_bulk.json](#)

```
POST /users/_bulk
{"index":{"_id":"10"}}
{"empId": "5e5a76285897070407bffd0b", "age": 29, "eyeColor": "brown", "name": "Conrad Norman",
"gender": "male", "company": "LOCAZONE", "email": "conradnorman@locazone.com", "phone": "+1 (836)
511-3994", "address": { "addId": "5e5a7628b8cbf7a261e08166", "door": 284, "street": "Ludlam Place",
"city": "Derwood", "state": "California", "pin": 6076 }, "favoriteFruit": "apple"}
```



```

{"index":{"_id":"11"}}
{"empId": "5e5a762821b4652df875a8d0", "age": 28, "eyeColor": "green", "name": "Atkinson Snider",
"gender": "male", "company": "OCTOCORE", "email": "atkinsonsnider@octocore.com", "phone": "+1 (903)
461-3255", "address": { "addId": "5e5a76281d3f5ab14ed92a87", "door": 315, "street": "Flatlands
Avenue", "city": "Baker", "state": "Ohio", "pin": 7405 }, "favoriteFruit": "apple"}
{"index":{"_id":"12"}}
{"empId": "5e5a76281fe5397a28480dcf", "age": 30, "eyeColor": "green", "name": "Katina Phelps", "gender":
"female", "company": "SENMEI", "email": "katinaphelps@senmei.com", "phone": "+1 (935) 554-3648",
"address": { "addId": "5e5a76282d085c36db741fec", "door": 507, "street": "Lewis Avenue", "city":
"Rote", "state": "Alaska", "pin": 2168 }, "favoriteFruit": "apple"}
{"index":{"_id":"13"}}
{"empId": "5e5a7628b0afdb763eb508ef", "age": 30, "eyeColor": "brown", "name": "Abbott Cline", "gender":
"male", "company": "COMTRACT", "email": "abbottcline@comtract.com", "phone": "+1 (966) 471-2316",
"address": { "addId": "5e5a76287c3dc6f68454b035", "door": 837, "street": "Keap Street", "city":
"Norvelt", "state": "Kentucky", "pin": 1947 }, "favoriteFruit": "apple"}
{"index":{"_id":"14"}}
{"empId": "5e5a762886f11fc1a5d1f289", "age": 39, "eyeColor": "green", "name": "Copeland Dejesus",
"gender": "male", "company": "UNI", "email": "copelanddejesus@uni.com", "phone": "+1 (940) 539-2249",
"address": { "addId": "5e5a76289eddff4d1accf774", "door": 479, "street": "Cheever Place", "city":
"Grandview", "state": "Maryland", "pin": 2808 }, "favoriteFruit": "apple"}
{"index":{"_id":"15"}}
{"empId": "5e5a7628b085cd6f052aedaa", "age": 40, "eyeColor": "brown", "name": "Debra Lewis", "gender":
"female", "company": "DOGTOWN", "email": "debralewis@dogtown.com", "phone": "+1 (985) 586-3140",
"address": { "addId": "5e5a7628a3bfce72565b3922", "door": 144, "street": "Fuller Place", "city":
"Frizzleburg", "state": "Federated States Of Micronesia", "pin": 830 }, "favoriteFruit": "apple"}
{"index":{"_id":"16"}}
{"empId": "5e5a762841294fa742351a3e", "age": 20, "eyeColor": "green", "name": "Martina Young",
"gender": "female", "company": "COASH", "email": "martinayoung@coash.com", "phone": "+1 (898) 595-
3459", "address": { "addId": "5e5a7628f3e5a2c30ce52893", "door": 652, "street": "Banker Street",
"city": "Diaperville", "state": "Northern Mariana Islands", "pin": 7552 }, "favoriteFruit": "apple"}
{"index":{"_id":"17"}}
{"empId": "5e5a762806a4c092f3cd157b", "age": 31, "eyeColor": "brown", "name": "Terra Goodman",
"gender": "female", "company": "TWIGGERY", "email": "terragoodman@twiggy.com", "phone": "+1 (845)
575-3415", "address": { "addId": "5e5a762856baa2032ada8be6", "door": 515, "street": "Horace Court",
"city": "Windsor", "state": "Connecticut", "pin": 9206 }, "favoriteFruit": "banana"}
{"index":{"_id":"18"}}
{"empId": "5e5a7628848dee3df1a5ef47", "age": 23, "eyeColor": "green", "name": "Hester Schroeder",
"gender": "female", "company": "SUNCLIPSE", "email": "hesterschroeder@sunclipse.com", "phone": "+1
(940) 500-3089", "address": { "addId": "5e5a762871308cdbc866f95a", "door": 951, "street": "Harden
Street", "city": "Ballico", "state": "Georgia", "pin": 8775 }, "favoriteFruit": "strawberry"}
{"index":{"_id":"19"}}
{"empId": "5e5a7628c5d9f93798b8f9df", "age": 24, "eyeColor": "blue", "name": "Jan Sherman", "gender":
"female", "company": "MIRACLIS", "email": "jansherman@miraclis.com", "phone": "+1 (986) 466-2510",
"address": { "addId": "5e5a7628b230d375ef79b9f9", "door": 631, "street": "Vandam Street", "city":
"Eagletown", "state": "Missouri", "pin": 7275 }, "favoriteFruit": "apple"}

```

Your output should look something like this:

```
{
  "took" : 31,
  "errors" : false,
  "items" : [
    {
      "index" : {
        "_index" : "users",
        "_type" : "_doc",
        "_id" : "10",
        "_version" : 1,
        "result" : "created",
        "_shards" : {
          "total" : 2,
          "successful" : 1,
          "failed" : 0
        },
        "_seq_no" : 6,
        "_primary_term" : 1,
        "status" : 201
      }
    },
    {
      "index" : {
        "_index" : "users",
```

Search all documents

GET /users/_search

```
{
  "query": {
    "match_all": {}
  }
}
```

Your output should look something like this:

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 11,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "users",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "empId" : "5e5a766afa73dddb81a1c939",
          "age" : 21,
          "eyeColor" : "brown",
          "name" : "Allison Strong",
          "gender" : "male",
          "company" : "ENERFORCE",
          "email" : "allisonstrong@enerforce.com",
          "phone" : "+1 (963) 503-2603",
          "address" : {
            "addId" : "5e5a766a0e58c8b905902b40",
            "door" : 822,
            "street" : "Bushwick Place",
            "city" : "Castleton",
            "state" : "Guam",
            "pin" : 2457
          },
          "favoriteFruit" : "kiwifruit"
        }
      },
      {
        "_index" : "users",
```

Search by query

GET /users/_search

```
{
  "query": {
    "match": {
      "gender": "male"
    }
  }
}
```

Your output should look like

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 5,
      "relation" : "eq"
    },
    "max_score" : 0.7801585,
    "hits" : [
      {
        "_index" : "users",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 0.7801585,
        "_source" : {
          "empId" : "5e5a766afa73dddb81a1c939",
          "age" : 21,
          "eyeColor" : "brown",
          "name" : "Allison Strong",
          "gender" : "male",
          "company" : "ENERFORCE",
          "email" : "allisonstrong@enerforce.com",
          "phone" : "+1 (963) 503-2603",
          "address" : {
            "addId" : "5e5a766a0e58c8b905902b40",
            "door" : 822,
            "street" : "Bushwick Place",
            "city" : "Castleton",
            "state" : "Guam",
            "pin" : 2457
          },
          "favoriteFruit" : "kiwifruit"
        }
      }
    ]
  }
}
```

Delete a Document

```
DELETE /users/_doc/1
```

Redis

Check the name of your container by executing the following command (it must be redis in our case)

```
docker ps
```

Login to your redis docker container using the following command

```
docker exec -it redis /bin/bash
```

You should see the following output

```
> docker exec -it redis /bin/bash
root@redis:/data#
```

Issue the following command to get into the redis-cli prompt

```
root@redis:/data# redis-cli
```

Issue the following command to get into the redis-cli prompt

```
root@redis:/data# redis-cli
127.0.0.1:6379>
```

Redis frequently used datatypes :

1. Lists
2. Sets
3. Hashes

Lists

Push to a List

```
RPUSH users Arun
```

```
127.0.0.1:6379> RPUSH users Arun
(integer) 1
```

Push multiple values to a List

```
RPUSH users Alice Daisy Jason
```

```
127.0.0.1:6379> RPUSH users Alice Daisy Jason
(integer) 4
```

Get values from a List

LRange users "Alice" "Daisy" "Jason"

```
127.0.0.1:6379> LRange users 0 -1  
1) "Arun"  
2) "Alice"  
3) "Daisy"  
4) "Jason"
```

Removing a value from a List

LPOP users

```
127.0.0.1:6379> LPOP users  
"Arun"
```

Removing a specific value from a List

LREM users 1 "Daisy"

```
127.0.0.1:6379> LREM users 1 "Daisy"  
(integer) 1  
127.0.0.1:6379> LRange users 0 -1  
1) "Alice"  
2) "Jason"
```

Find the length of the List

LLEN users

```
127.0.0.1:6379> LLEN users  
(integer) 2
```

Sets

Push to a Set

SADD books "machine learning"

```
127.0.0.1:6379> SADD books "machine learning"  
(integer) 1
```

Push multiple values to a Set

SADD books "machine learning" "microservices" "data science" "machine learning"

```
127.0.0.1:6379> SADD books "machine learning" "microservices" "data science" "machine learning"
(integer) 2
```

Listing members in a Set

```
SMEMBERS books
```

```
127.0.0.1:6379> SMEMBERS books
1) "data science"
2) "microservices"
3) "machine learning"
```

Removing a value from a Set

```
SREM books "microservices"
```

```
127.0.0.1:6379> SREM books "microservices"
(integer) 1
```

Checking membership in a Set

```
SISMEMBER books "machine learning"
SISMEMBER books "no book"
```

```
127.0.0.1:6379> SISMEMBER books "machine learning"
(integer) 1
127.0.0.1:6379> SISMEMBER books "no book"
(integer) 0
```

Hashes

Push to a Hash

```
HSET phones "apple" 11
HSET phones "samsung" 25
```

HSET phones "LG" 16

```
127.0.0.1:6379> HSET phones "apple" 11
(integer) 1
127.0.0.1:6379> HSET phones "samsung" 25
(integer) 1
127.0.0.1:6379> HSET phones "LG" 16
(integer) 1
```

Pushing multiple values to a Hash

HMSET phones "Oppo" 14 "Huawei" 17 "Redmi" 29

```
127.0.0.1:6379> HMSET phones "Oppo" 14 "Huawei" 17 "Redmi" 29
OK
```

Get value for a key from a Hash

HGET phones "LG"

```
127.0.0.1:6379> HGET phones "LG"
"16"
```

Get all values from a Hash

HGETALL phones


```
127.0.0.1:6379> HGETALL phones
1) "apple"
2) "11"
3) "samsung"
4) "25"
5) "LG"
6) "16"
7) "Oppo"
8) "14"
9) "Huawei"
10) "17"
11) "Redmi"
12) "29"
```

Removing a value from a Hash

```
HDEL phones "Oppo"
```

```
127.0.0.1:6379> HDEL phones "Oppo"
(integer) 1
127.0.0.1:6379> HGETALL phones
1) "apple"
2) "11"
3) "samsung"
4) "25"
5) "LG"
6) "16"
7) "Huawei"
8) "17"
9) "Redmi"
10) "29"
```

Neo4J

Open up Neo4J browser

by pointing your browser to

```
http://localhost:7474
```

Create new Nodes

```
CREATE (alice:Person {name:"Alice Wells", gender:"female", age:36})
CREATE (jeremy:Person {name:"Jeremy Smith", gender:"male", age:40})
```

The screenshot shows the Neo4j Browser interface. At the top, there's a code editor with two lines of Cypher: `1 CREATE (alice:User {name:"Alice Wells", gender: "female", age:36})` and `2 CREATE (jeremy:User {name:"Jeremy Smith", gender: "male", age:40})`. Below the editor is a blue banner with the text "To enjoy the full Neo4j Browser experience, we advise you to use [Neo4j Browser Sync](#)". The execution results show the command `neo4j$ CREATE (alice:User {name:"Alice Wells", gender: "female", age:36}) CREATE (jeremy:User {name:"Jeremy Smith", gender: "male", age:40})` and a message: "Added 2 labels, created 2 nodes, set 6 properties, completed after 79 ms." On the left sidebar, the "Table" view is selected.

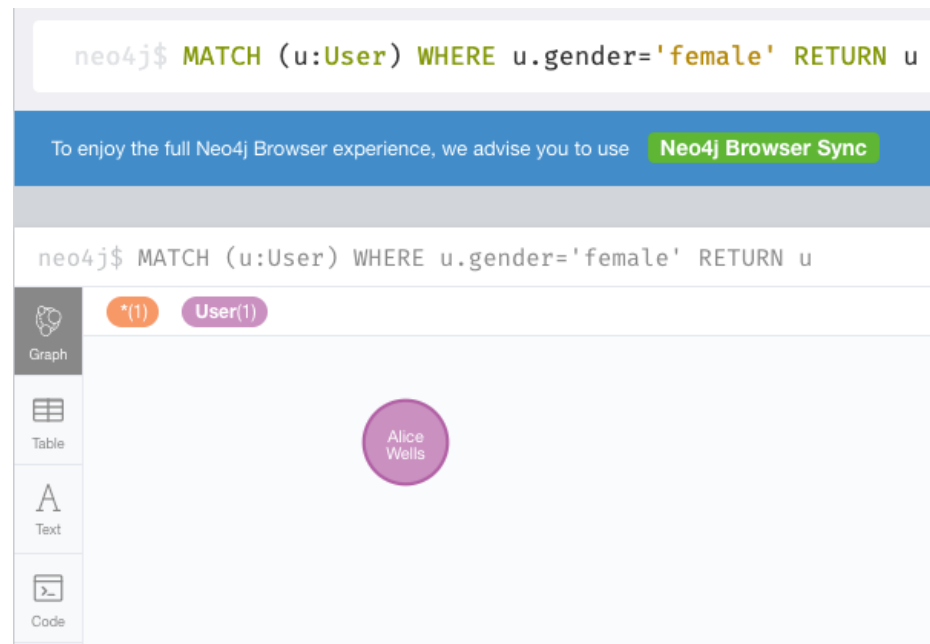
Create new Relationship

```
MATCH (a:Person), (j:Person) WHERE a.name="Alice Wells" AND j.name="Jeremy Smith"
CREATE (a)-[:Married]->(j)
```

The screenshot shows the Neo4j Browser interface. The code editor contains: `1 MATCH (a:User), (j:User) WHERE a.name="Alice Wells" AND j.name="Jeremy Smith"` and `2 CREATE (a)-[:Married]->(j)`. Below the editor is the same blue banner: "To enjoy the full Neo4j Browser experience, we advise you to use [Neo4j Browser Sync](#)". The execution results show the command `neo4j$ MATCH (a:User), (j:User) WHERE a.name="Alice Wells" AND j.name="Jeremy Smith" CREATE (a)-[:Married]->(j)` and a message: "Created 1 relationship, completed after 2 ms." On the left sidebar, the "Table" view is selected.

Query a single node

```
MATCH (u:Person) WHERE u.gender='female' RETURN u
```



Get all nodes and relationships

```
MATCH (n1)-[r]->(n2) RETURN r, n1, n2
```

(or)

```
MATCH (n) RETURN n
```

```
1 // Get some data
2 MATCH (n1)-[r]→(n2) RETURN r, n1, n2
```

To enjoy the full Neo4j Browser experience, we advise you to use [Neo4j Browser Sync](#)

```
neo4j$ MATCH (n1)-[r]→(n2) RETURN r, n1, n2
```

Graph

* (2) User(2)

* (1) Married(1)

Table

Text

Code

The graph shows two nodes: 'Alice Wells' and 'Jeremy Smith', both represented as purple circles. A directed relationship labeled 'Married' connects 'Alice Wells' to 'Jeremy Smith'.

Deleting a node

(Relationship has to be deleted first before deleting node)

```
MATCH (n1)-[r]→(n2) DELETE r
```

Delete Relationship

```
neo4j$ MATCH (n1)-[r]→(n2) DELETE r
```

To enjoy the full Neo4j Browser experience, we advise you to use [Neo4j Browser Sync](#)

```
neo4j$ MATCH (n1)-[r]→(n2) DELETE r
```

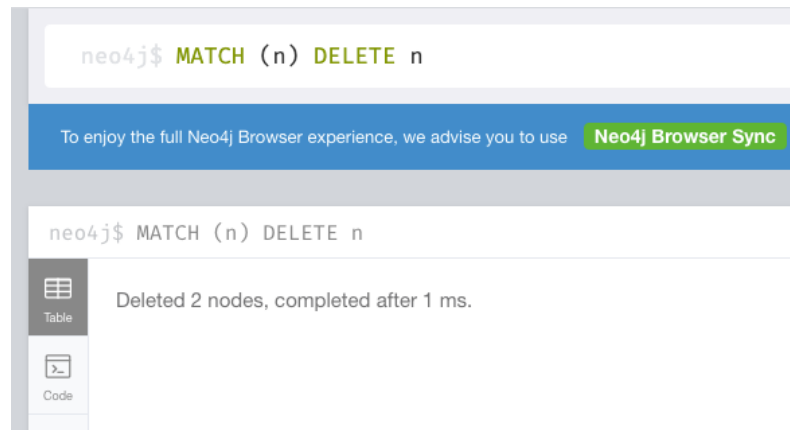
Table

Code

Deleted 2 relationships, completed after 3 ms.

Delete Nodes

```
MATCH (n) DELETE n
```



Load Graph from CSV

Copy the users.csv from data-management/neo4j directory into the neo4j docker container.

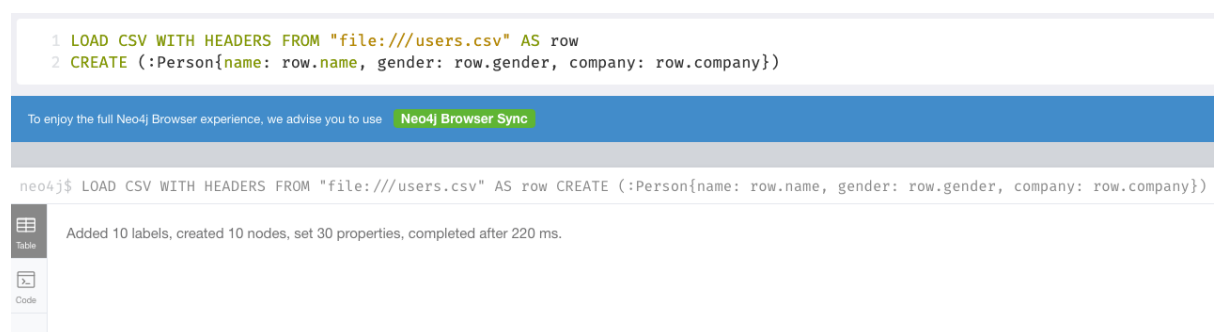
Issue the following command from your terminal

```
docker cp neo4j/users.csv neo4j:/var/lib/neo4j/import/users.csv
```

Issue the following command from your Neo4J browser window

Load CSV

```
LOAD CSV WITH HEADERS FROM "file:///users.csv" AS row
CREATE (:Person{name: row.name, gender: row.gender, company: row.company})
```



neo4j\$ MATCH (n) RETURN n

To enjoy the full Neo4j Browser experience, we advise you to use [Neo4j Browser Sync](#)

neo4j\$ MATCH (n) RETURN n

Create Nodes and Relationships

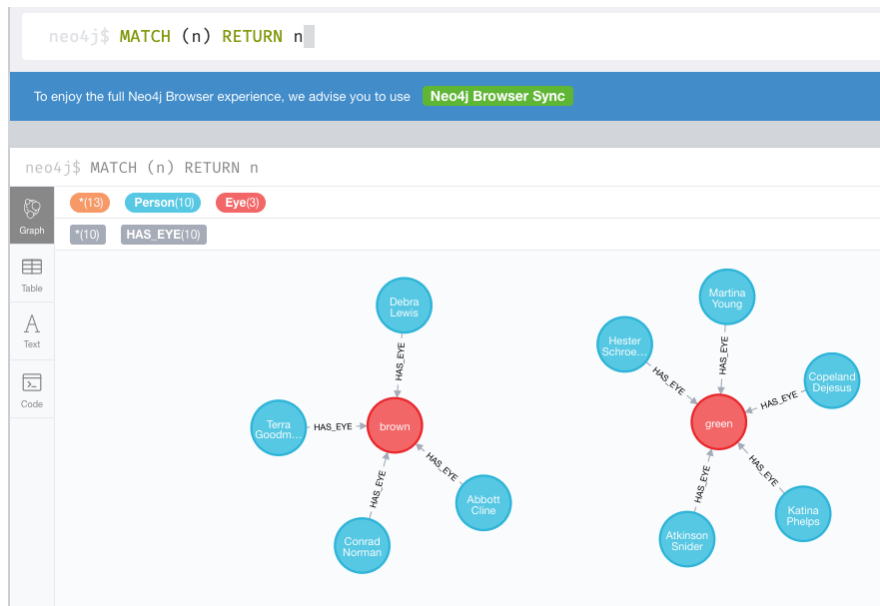
```
LOAD CSV WITH HEADERS FROM "file:///users.csv" AS row
CREATE (person:Person{name: row.name, gender: row.gender, company: row.company})
MERGE (eye:Eye{color: row.eyeColor})
CREATE (person)-[:HAS_EYE]->(eye)
```

```
1 LOAD CSV WITH HEADERS FROM "file:///users.csv" AS row
2 CREATE (person:Person{name: row.name, gender: row.gender, company: row.company})
3 MERGE (eye:Eye{color: row.eyeColor})
4 CREATE (person)-[:HAS_EYE]->(eye)
```

To enjoy the full Neo4j Browser experience, we advise you to use [Neo4j Browser Sync](#)

neo4j\$ LOAD CSV WITH HEADERS FROM "file:///users.csv" AS row CREATE (person:Person{name: row.name, gender: row.gender, company: row.company})

Added 10 labels, created 10 nodes, set 30 properties, created 10 relationships, completed after 19 ms.



MongoDB

Check the name of your container by executing the following command (it must be mongodb in our case)

```
docker ps
```

Login to your mongodb docker container using the following command

```
docker exec -it mongodb /bin/bash
```

You should see the following output

```
> docker exec -it mongodb /bin/bash
root@mongodb:/#
```

Issue the following command to get into the mongo prompt

```
mongo --username root --password
```

Password is **rootpassword** as specified in your docker-compose.yml

```
root@mongodb:/# mongo --username root --password
MongoDB shell version v4.2.3
Enter password:
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("60e37399-3bec-449c-a989-e6680ab3f416") }
MongoDB server version: 4.2.3
Server has startup warnings:
2020-03-01T02:39:59.474+0000 I STORAGE [initandlisten]
2020-03-01T02:39:59.474+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem
```

Create/Switch to a database

```
use usersdb
```

Create a new collection

```
db.createCollection("users")
```

```
> db.createCollection("users")
{ "ok" : 1 }
```

Insert a new document

You can get a sample document from [data-management/mongo/users_single.json](#)

```
db.users.insert({ "empId": "5e5a766afa73dddb81a1c939", "age": 21, "eyeColor": "brown", "name": "Allison Strong", "gender": "male", "company": "ENERFORCE", "email": "allisonstrong@enerforce.com", "phone": "+1 (963) 503-2603", "address": { "addId": "5e5a766a0e58c8b905902b40", "door": 822, "street": "Bushwick Place", "city": "Castleton", "state": "Guam", "pin": 2457 }, "favoriteFruit": "banana" })
```

(Insert is a function call. Notice the parenthesis before the JSON argument)

```
> db.users.insert({ "empId": "5e5a766afa73dddb81a1c939", "age": 21, "eyeColor": "brown", "name": "Allison Strong", "gender": "male", "company": "ENERFORCE", "email": "allisonstrong@enerforce.com", "phone": "+1 (963) 503-2603", "address": { "addId": "5e5a766a0e58c8b905902b40", "door": 822, "street": "Bushwick Place", "city": "Castleton", "state": "Guam", "pin": 2457 }, "favoriteFruit": "banana" })
WriteResult({ "nInserted" : 1 })
```

Bulk insert documents

Bulk insertion is done by passing an array into the insert function. You can get a sample document from [data-management/mongo/users_bulk.json](#)

```
db.users.insert([
... { "_id": "10", "empId": "5e5a76285897070407bffd0b", "age": 29, "eyeColor": "brown", "name": "Conrad Norman", "gender": "male", "company": "LOCAZONE", "email": "conradnorman@locazone.com", "phone": "+1 (836) 511-3994", "address": { "addId": "5e5a7628b8cbf7a261e08166", "door": 284, "street": "Ludlam Place", "city": "Derwood", "state": "California", "pin": 6076 }, "favoriteFruit": "apple"},
```



```

...{"_id":"11","empld":"5e5a762821b4652df875a8d0","age":28,"eyeColor":"green","name":"Atkinson
Snider","gender":"male","company":"OCTOCORE","email":"atkinsonsnider@octocore.com","phone":
"+1(903)461-3255","address":{"addId":"5e5a76281d3f5ab14ed92a87","door":315,"street":
"Flatlands Avenue","city":"Baker","state":"Ohio","pin":7405},"favoriteFruit":"apple"},
...{"_id":"12","empld":"5e5a76281fe5397a28480dcf","age":30,"eyeColor":"green","name":"Katina
Phelps","gender":"female","company":"SENMEI","email":"katinaphelps@senmei.com","phone":"+1
(935)554-3648","address":{"addId":"5e5a76282d085c36db741fec","door":507,"street":"Lewis
Avenue","city":"Rote","state":"Alaska","pin":2168},"favoriteFruit":"apple"},
...{"_id":"13","empld":"5e5a7628b0afdb763eb508ef","age":30,"eyeColor":"brown","name":"Abbott
Cline","gender":"male","company":"COMTRACT","email":"abbottcline@contract.com","phone":"+1
(966)471-2316","address":{"addId":"5e5a76287c3dc6f68454b035","door":837,"street":"Keap
Street","city":"Norvelt","state":"Kentucky","pin":1947},"favoriteFruit":"apple"},
...{"_id":"14","empld":"5e5a762886f11fc1a5d1f289","age":39,"eyeColor":"green","name":"Copeland
Dejesus","gender":"male","company":"UNI","email":"copelanddejesus@uni.com","phone":"+1(940)
539-2249","address":{"addId":"5e5a76289eddf4d1accf774","door":479,"street":"Cheever Place",
"city":"Grandview","state":"Maryland","pin":2808},"favoriteFruit":"apple"},
...{"_id":"15","empld":"5e5a7628b085cd6f052aedaa","age":40,"eyeColor":"brown","name":"Debra
Lewis","gender":"female","company":"DOGTOWN","email":"debralewis@dogtown.com","phone":"+1
(985)586-3140","address":{"addId":"5e5a7628a3bfce72565b3922","door":144,"street":"Fuller
Place","city":"Frizzleburg","state":"Federated States Of Micronesia","pin":830},"favoriteFruit":
"apple"},
...{"_id":"16","empld":"5e5a762841294fa742351a3e","age":20,"eyeColor":"green","name":"Martina
Young","gender":"female","company":"COASH","email":"martinayoung@coash.com","phone":"+1
(898)595-3459","address":{"addId":"5e5a7628f3e5a2c30ce52893","door":652,"street":"Banker
Street","city":"Diaperville","state":"Northern Mariana Islands","pin":7552},"favoriteFruit":"apple"},
...{"_id":"17","empld":"5e5a762806a4c092f3cd157b","age":31,"eyeColor":"brown","name":"Terra
Goodman","gender":"female","company":"TWIGGERY","email":"terragoodman@twiggery.com",
"phone":"+1(845)575-3415","address":{"addId":"5e5a762856baa2032ada8be6","door":515,
"street":"Horace Court","city":"Windsor","state":"Connecticut","pin":9206},"favoriteFruit":
"banana"},
...{"_id":"18","empld":"5e5a7628848dee3df1a5ef47","age":23,"eyeColor":"green","name":"Hester
Schroeder","gender":"female","company":"SUNCLIPSE","email":"hesterschroeder@sunclipse.com",
"phone":"+1(940)500-3089","address":{"addId":"5e5a762871308cdbc866f95a","door":951,
"street":"Harden Street","city":"Ballico","state":"Georgia","pin":8775},"favoriteFruit":
"strawberry"},
...{"_id":"19","empld":"5e5a7628c5d9f93798b8f9df","age":24,"eyeColor":"blue","name":"Jan
Sherman","gender":"female","company":"MIRACLIS","email":"jansherman@miraclis.com","phone":"+1
(986)466-2510","address":{"addId":"5e5a7628b230d375ef79b9f9","door":631,"street":"Vandam
Street","city":"Eagletown","state":"Missouri","pin":7275},"favoriteFruit":"apple"},
...])

```

```
... {"_id": "19", "empId": "5e5a7628c5d9f93798b8f9df", "
  "company": "MIRACLIS", "email": "jansherman@miraclis.c
375ef79b9f9", "door": 631, "street": "Vandam Stree
oriteFruit": "apple"},
... ])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 10,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

Query all documents in a collection

```
db.users.find()
> db.users.find()
{ "_id" : ObjectId("5e5b2577d48d8c4f88ae0c9c"), "empId" :
lison Strong", "gender" : "male", "company" : "ENERFORCE",
address" : { "addId" : "5e5a766a0e58c8b905902b40", "door"
, "pin" : 2457 }, "favoriteFruit" : "banana" }
{ "_id" : "10", "empId" : "5e5a76285897070407bffd0b", "age
"company" : "LOCAZONE", "email" : "conradnorman@locazone.
f7a261e08166", "door" : 284, "street" : "Ludlam Place", "c
: "apple" }
```

Find document in a collection matching a criteria

```
db.users.find({"eyeColor": "blue"})
> db.users.find({"eyeColor": "blue"})
{ "_id" : "19", "empId" : "5e5a7628c5d9f93798b8f9df", "age" : 24, "
"company" : "MIRACLIS", "email" : "jansherman@miraclis.com", "phone
5ef79b9f9", "door" : 631, "street" : "Vandam Street", "city" : "Eag
"apple" }
```

Find document in a collection matching multiple criteria

With regular expression matching

```
#Second condition is equivalent to "Debra%"
db.users.find({$and:[{"eyeColor": "brown"}, {"name": /^Debra/}]})

#Second condition is equivalent to "%Lewis"
db.users.find({$and:[{"eyeColor": "brown"}, {"name": /Lewis$/}]})
```

```
#Second condition is equivalent to "%Lew%"  
db.users.find({$and:[{"eyeColor": "brown"}, {"name": /Lew/}]})
```

```
> db.users.find({$and:[{"eyeColor": "brown"}, {"name": /^Debra/}]})  
{ "_id" : "15", "empId" : "5e5a7628b085cd6f052aadaa", "age" : 40,  
  "company" : "DOGTOWN", "email" : "debralewis@dogtown.com", "phone  
565b3922", "door" : 144, "street" : "Fuller Place", "city" : "Friz  
, "favoriteFruit" : "apple" }
```

Removing a document

```
db.users.remove({"eyeColor": "blue"})
```

```
> db.users.remove({"eyeColor": "blue"})  
WriteResult({ "nRemoved" : 1 })  
> db.users.find({"eyeColor": "blue"})
```

Updating a document

```
db.users.update({"name": "Debra Lewis"}, {$set: {"favoriteFruit": "kiwi"}})
```

```
> db.users.update({"name": "Debra Lewis"}, {$set: {"favoriteFruit": "kiwi"}})  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.users.find({"name": "Debra Lewis"})  
{ "_id" : "15", "empId" : "5e5a7628b085cd6f052aadaa", "age" : 40, "eyeColor" :  
, "company" : "DOGTOWN", "email" : "debralewis@dogtown.com", "phone" : "+1 (985  
72565b3922", "door" : 144, "street" : "Fuller Place", "city" : "Frizzleburg", "  
30 }, "favoriteFruit" : "kiwi" }
```

End of workshop