



MongoDB/Elasticsearch Setup Day!

1. Set up MongoDB on your local or EC2 machine
 - A. Install the software
 - B. Initialize a database on the command line
 - C. Start mongod, the MongoDB server
 - D. Interact with it using mongo

2. Set up Elasticsearch on your local or EC2 machine
 - A. Install the software
 - B. Create Elasticsearch directories/folders
 - C. Start elasticsearch, the Elasticsearch server
 - D. Interact with it using web service client such as curl or Postman

1A. Install MongoDB

- As before, we have different machines, platforms, and comfort zones, so go here and find the best fit for you:

<https://docs.mongodb.com/guides/server/install/>

- For EC2 installation, you can try the Linux tab above or these Amazon Linux-specific instructions:

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-amazon/>

1B. Initialize the MongoDB database

- How about this for initialization: MongoDB only needs a directory that the server process can read and write—just create it before starting the server
- File creation within that directory is done automatically when the server is started
- Their installation guide has more detail, if needed:

<https://docs.mongodb.com/guides/server/install/>

1C. Start mongod, the MongoDB server

- Although the MongoDB server will default to /data/db for its database location, it's useful to get into the habit of specifying one directly when you start the server:

```
mongod --dbpath location-of-database
```

- To stop it, just hit Ctrl-C on your keyboard
- This is, of course, the minimal approach—options abound for production use

1D. Run mongo

- mongo is MongoDB's built-in command-line client:

```
mongo mongodb://localhost
```

- As with most database systems, there are lots more options available—we're just using the most expedient one so that we can get to work right away
- Because of this, don't be surprised if you get a bunch of warnings regarding authentication and connectivity—that's expected for a quick setup like this

2A. Install Elasticsearch

- Start here for Elasticsearch—pick a download option from the top, then follow the steps at the bottom:

<https://www.elastic.co/downloads/elasticsearch>

- More detailed steps can be found here:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html>

2B. Create Elasticsearch folders

This one looks more complicated than documentation you may see online, but this makes things more explicit:

1. Decide where you want your data to live and where you want log files to live
2. Create one empty folder/directory for each—you will name them when you start the server
3. Other folders may be involved depending on how you installed Elasticsearch—we'll handle that case-to-case

2C. Start the elasticsearch server

- Although the Elasticsearch server can read from a configuration file, we're opting to state data and log locations when you start the server:

```
elasticsearch -Epath.data=location-of-database  
              -Epath.logs=location-of-log-files
```

- Some installations require absolute paths here, so keep that in mind in case you encounter odd errors
- To stop it, just hit Ctrl-C on your keyboard

2D. It's a web service!

- Elasticsearch doesn't have a built-in command-line interface—instead, it is a “pure” web service that can be reached at default port 9200, so you interact with it using something like curl or Postman
- The “Getting Started” site introduces you to some endpoints after the installation step:

<https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html>

mongo Basics

A MongoDB server hosts zero or more databases, and each database contains zero or more collections:

- `show dbs` lists a server's databases
- use `db-name` makes `db-name` the current database
- `show collections` lists the current DB's collections
- `db.collection-name.func` will invoke one of many collection methods—this is effectively MongoDB's “query language”

MongoDB “auto-creates” things—if you use `db-name` and `db-name` doesn't exist yet, `db-name` will be created; if you invoke `db.collection-name.func` and `collection-name` doesn't exist yet, it will be created

- `help` lists some basic commands as well as how to get more help
- `exit` or Control-D will get you out of `mongo`

The “CRUD Guides” collection in MongoDB's Guides section provides a decent introductory walkthrough for interacting with MongoDB's collections:

<https://docs.mongodb.com/guides/>

Elasticsearch Basics

- In its default configuration, Elasticsearch can be reached at port 9200—so if it's running on your machine, you would use <http://localhost:9200>
- Elasticsearch presents itself as a set of indices—this is equivalent to a MongoDB collection
- To interact with an index, you would append its name at the beginning of the endpoint; for example, if you have a “movie” index, endpoints start with:

<http://localhost:9200/movie>

- Like MongoDB, an index auto-creates itself once you start POSTing or PUTting documents there
- An index can have `/mappings`—essentially, a schema
- Invoking DELETE on the index URL will wipe it out—use with care! (but handy when you're experimenting)
- Endpoint segments with underscores represent various activities: `_cat`, `_doc`, `_search`, `_bulk`, to name a few
- Payloads are provided as JSON—thus, Elasticsearch queries are all JSON objects with a specific structure, and results are returned as JSON as well
- In many ways, learning Elasticsearch at this level is a lot like learning a web service (for better or for worse!)