OFFICIAL REPOSITORY

mongo (/r/_/mongo/) ☆

Last pushed: 4 hours ago

Repo Info (/_/mongo/)

Short Description

MongoDB document databases provide high availability and easy scalability.

Full Description

Note: the description for this image is longer than the Hub length limit of 25000, so has been trimmed. The full description can be found at https://github.com/docker-library/docs/tree/master/mongo/README.md (https://github.com/docker-library/docs/tree/master/mongo/README.md). See docker/hub-beta-feedback#238 (https://github.com/docker/hub-beta-feedback/issues/238) for more information.

Supported tags and respective `Dockerfile` links

Simple Tags

- `3.2.21-jessie`, `3.2-jessie` (*3.2/Dockerfile*) (https://github.com/docker-library/mongo/blob/6932ac255d29759af9a74c6931faeb02de0fe53e/3.2/Dockerfile)
- `3.2.21-windowsservercore-ltsc2016`, `3.2-windowsservercore-ltsc2016` (*3.2/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/32e5645325ae5dff2ae6f934049cbd6b4635aec0/3.2/windows/windowsservercore-ltsc2016/Dockerfile)
- `3.2.21-windowsservercore-1709`, `3.2-windowsservercore-1709` (*3.2/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/32e5645325ae5dff2ae6f934049cbd6b4635aec0/3.2/windows/windowsservercore-1709/Dockerfile)
- `3.4.18-jessie`, `3.4-jessie` (*3.4/Dockerfile*) (https://github.com/docker-library/mongo/blob/6932ac255d29759af9a74c6931faeb02de0fe53e/3.4/Dockerfile)
- `3.4.18-windowsservercore-ltsc2016`, `3.4-windowsservercore-ltsc2016` (*3.4/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/df76a27fa96398f234a2287d327687e37d3c8d98/3.4/windows/windowsservercore-ltsc2016/Dockerfile)
- `3.4.18-windowsservercore-1709`, `3.4-windowsservercore-1709` (*3.4/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/df76a27fa96398f234a2287d327687e37d3c8d98/3.4/windows/windowsservercore-1709/Dockerfile)
- `3.6.9-stretch`, `3.6-stretch`, `3-stretch` (*3.6/Dockerfile*) (https://github.com/docker-library/mongo/blob/cb722bcc413b6789064d874b5231f9e23cc81540/3.6/Dockerfile)
- `3.6.9-windowsservercore-ltsc2016`, `3.6-windowsservercore-ltsc2016`, `3-windowsservercore-ltsc2016` (*3.6/windows/windowsservercore-ltsc2016/Dockerfile*)

(https://github.com/docker-library/mongo/blob/2674c80f0674f29484c4ccb4aff7d076227d5bc9/3.6/windows/windowsservercore-ltsc2016/Dockerfile)

- `3.6.9-windowsservercore-1709`, `3.6-windowsservercore-1709`, `3-windowsservercore-1709` (*3.6/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/2674c80f0674f29484c4ccb4aff7d076227d5bc9/3.6/windows/windowsservercore-1709/Dockerfile)

- `4.0.4-xenial`, `4.0-xenial`, `4-xenial`, `xenial` (*4.0/Dockerfile*) (https://github.com/docker-library/mongo/blob/6932ac255d29759af9a74c6931faeb02de0fe53e/4.0/Dockerfile)

- `4.0.4-windowsservercore-ltsc2016`, `4.0-windowsservercore-ltsc2016`, `4-windowsservercore-ltsc2016`, `windowsservercore-ltsc2016` (*4.0/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/b1c209cdd9b890cd62e85a2c7055fb8f00c7d248/4.0/windows/windowsservercore-ltsc2016/Dockerfile)

- `4.0.4-windowsservercore-1709`, `4.0-windowsservercore-1709`, `4-windowsservercore-1709`, `windowsservercore-1709` (*4.0/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/b1c209cdd9b890cd62e85a2c7055fb8f00c7d248/4.0/windows/windowsservercore-1709/Dockerfile)

- `4.0.4-windowsservercore-1803`, `4.0-windowsservercore-1803`, `4-windowsservercore-1803`, `windowsservercore-1803` (*4.0/windows/windowsservercore-1803/Dockerfile*) (https://github.com/docker-library/mongo/blob/b1c209cdd9b890cd62e85a2c7055fb8f00c7d248/4.0/windows/windowsservercore-1803/Dockerfile)

- `4.1.5-xenial`, `4.1-xenial`, `unstable-xenial` (*4.1/Dockerfile*) (https://github.com/docker-library/mongo/blob/6932ac255d29759af9a74c6931faeb02de0fe53e/4.1/Dockerfile)

- `4.1.5-windowsservercore-ltsc2016`, `4.1-windowsservercore-ltsc2016`, `unstable-windowsservercore-ltsc2016` (*4.1/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/d5e810eda79184ce146144a18105c3e5ab2130b7/4.1/windows/windowsservercore-ltsc2016/Dockerfile)

- `4.1.5-windowsservercore-1709`, `4.1-windowsservercore-1709`, `unstable-windowsservercore-1709` (*4.1/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/d5e810eda79184ce146144a18105c3e5ab2130b7/4.1/windows/windowsservercore-1709/Dockerfile)

- `4.1.5-windowsservercore-1803`, `4.1-windowsservercore-1803`, `unstable-windowsservercore-1803` (*4.1/windows/windowsservercore-1803/Dockerfile*) (https://github.com/docker-library/mongo/blob/d5e810eda79184ce146144a18105c3e5ab2130b7/4.1/windows/windowsservercore-1803/Dockerfile)

Shared Tags

- `3.2.21`, `3.2`:
  - `3.2.21-jessie` (*3.2/Dockerfile*) (https://github.com/docker-library/mongo/blob/6932ac255d29759af9a74c6931faeb02de0fe53e/3.2/Dockerfile)
  - `3.2.21-windowsservercore-ltsc2016` (*3.2/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/32e5645325ae5dff2ae6f934049cbd6b4635aec0/3.2/windows/windowsservercore-ltsc2016/Dockerfile)

- `3.2.21-windowsservercore-1709` (*3.2/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/32e5645325ae5dff2ae6f934049cbd6b4635aec0/3.2/windows/windowsservercore-1709/Dockerfile)
- `3.2.21-windowsservercore` , `3.2-windowsservercore` :
  - `3.2.21-windowsservercore-ltsc2016` (*3.2/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/32e5645325ae5dff2ae6f934049cbd6b4635aec0/3.2/windows/windowsservercore-ltsc2016/Dockerfile)
  - `3.2.21-windowsservercore-1709` (*3.2/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/32e5645325ae5dff2ae6f934049cbd6b4635aec0/3.2/windows/windowsservercore-1709/Dockerfile)
- `3.4.18` , `3.4` :
  - `3.4.18-jessie` (*3.4/Dockerfile*) (https://github.com/docker-library/mongo/blob/6932ac255d29759af9a74c6931faeb02de0fe53e/3.4/Dockerfile)
  - `3.4.18-windowsservercore-ltsc2016` (*3.4/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/df76a27fa96398f234a2287d327687e37d3c8d98/3.4/windows/windowsservercore-ltsc2016/Dockerfile)
  - `3.4.18-windowsservercore-1709` (*3.4/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/df76a27fa96398f234a2287d327687e37d3c8d98/3.4/windows/windowsservercore-1709/Dockerfile)
- `3.4.18-windowsservercore` , `3.4-windowsservercore` :
  - `3.4.18-windowsservercore-ltsc2016` (*3.4/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/df76a27fa96398f234a2287d327687e37d3c8d98/3.4/windows/windowsservercore-ltsc2016/Dockerfile)
  - `3.4.18-windowsservercore-1709` (*3.4/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/df76a27fa96398f234a2287d327687e37d3c8d98/3.4/windows/windowsservercore-1709/Dockerfile)
- `3.6.9` , `3.6` , `3` :
  - `3.6.9-stretch` (*3.6/Dockerfile*) (https://github.com/docker-library/mongo/blob/cb722bcc413b6789064d874b5231f9e23cc81540/3.6/Dockerfile)
  - `3.6.9-windowsservercore-ltsc2016` (*3.6/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/2674c80f0674f29484c4ccb4aff7d076227d5bc9/3.6/windows/windowsservercore-ltsc2016/Dockerfile)
  - `3.6.9-windowsservercore-1709` (*3.6/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/2674c80f0674f29484c4ccb4aff7d076227d5bc9/3.6/windows/windowsservercore-1709/Dockerfile)
- `3.6.9-windowsservercore` , `3.6-windowsservercore` , `3-windowsservercore` :
  - `3.6.9-windowsservercore-ltsc2016` (*3.6/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/2674c80f0674f29484c4ccb4aff7d076227d5bc9/3.6/windows/windowsservercore-ltsc2016/Dockerfile)

- `3.6.9-windowsservercore-1709` (*3.6/windows/windowsservercore-1709/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/2674c80f0674f29484c4ccb4aff7d076227d5bc9/3.6/windows/windowsservercore-
    1709/Dockerfile)
- `4.0.4`, `4.0`, `4`, `latest`:
  - `4.0.4-xenial` (*4.0/Dockerfile*) (https://github.com/docker-
    library/mongo/blob/6932ac255d29759af9a74c6931faeb02de0fe53e/4.0/Dockerfile)
  - `4.0.4-windowsservercore-ltsc2016` (*4.0/windows/windowsservercore-ltsc2016/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/b1c209cdd9b890cd62e85a2c7055fb8f00c7d248/4.0/windows/windowsservercore-
    ltsc2016/Dockerfile)
  - `4.0.4-windowsservercore-1709` (*4.0/windows/windowsservercore-1709/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/b1c209cdd9b890cd62e85a2c7055fb8f00c7d248/4.0/windows/windowsservercore-
    1709/Dockerfile)
  - `4.0.4-windowsservercore-1803` (*4.0/windows/windowsservercore-1803/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/b1c209cdd9b890cd62e85a2c7055fb8f00c7d248/4.0/windows/windowsservercore-
    1803/Dockerfile)
- `4.0.4-windowsservercore`, `4.0-windowsservercore`, `4-windowsservercore`,
  `windowsservercore`:
  - `4.0.4-windowsservercore-ltsc2016` (*4.0/windows/windowsservercore-ltsc2016/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/b1c209cdd9b890cd62e85a2c7055fb8f00c7d248/4.0/windows/windowsservercore-
    ltsc2016/Dockerfile)
  - `4.0.4-windowsservercore-1709` (*4.0/windows/windowsservercore-1709/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/b1c209cdd9b890cd62e85a2c7055fb8f00c7d248/4.0/windows/windowsservercore-
    1709/Dockerfile)
  - `4.0.4-windowsservercore-1803` (*4.0/windows/windowsservercore-1803/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/b1c209cdd9b890cd62e85a2c7055fb8f00c7d248/4.0/windows/windowsservercore-
    1803/Dockerfile)
- `4.1.5`, `4.1`, `unstable`:
  - `4.1.5-xenial` (*4.1/Dockerfile*) (https://github.com/docker-
    library/mongo/blob/6932ac255d29759af9a74c6931faeb02de0fe53e/4.1/Dockerfile)
  - `4.1.5-windowsservercore-ltsc2016` (*4.1/windows/windowsservercore-ltsc2016/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/d5e810eda79184ce146144a18105c3e5ab2130b7/4.1/windows/windowsservercore-
    ltsc2016/Dockerfile)
  - `4.1.5-windowsservercore-1709` (*4.1/windows/windowsservercore-1709/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/d5e810eda79184ce146144a18105c3e5ab2130b7/4.1/windows/windowsservercore-
    1709/Dockerfile)
  - `4.1.5-windowsservercore-1803` (*4.1/windows/windowsservercore-1803/Dockerfile*)
    (https://github.com/docker-
    library/mongo/blob/d5e810eda79184ce146144a18105c3e5ab2130b7/4.1/windows/windowsservercore-
    1803/Dockerfile)
- `4.1.5-windowsservercore`, `4.1-windowsservercore`, `unstable-windowsservercore`:

- `4.1.5-windowsservercore-ltsc2016` (*4.1/windows/windowsservercore-ltsc2016/Dockerfile*) (https://github.com/docker-library/mongo/blob/d5e810eda79184ce146144a18105c3e5ab2130b7/4.1/windows/windowsservercore-ltsc2016/Dockerfile)
  - `4.1.5-windowsservercore-1709` (*4.1/windows/windowsservercore-1709/Dockerfile*) (https://github.com/docker-library/mongo/blob/d5e810eda79184ce146144a18105c3e5ab2130b7/4.1/windows/windowsservercore-1709/Dockerfile)
  - `4.1.5-windowsservercore-1803` (*4.1/windows/windowsservercore-1803/Dockerfile*) (https://github.com/docker-library/mongo/blob/d5e810eda79184ce146144a18105c3e5ab2130b7/4.1/windows/windowsservercore-1803/Dockerfile)

Quick reference

- **Where to get help**:
  the Docker Community Forums (https://forums.docker.com/), the Docker Community Slack (https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/), or Stack Overflow (https://stackoverflow.com/search?tab=newest&q=docker)

- **Where to file issues**:
  https://github.com/docker-library/mongo/issues (https://github.com/docker-library/mongo/issues)

- **Maintained by**:
  the Docker Community (https://github.com/docker-library/mongo)

- **Supported architectures**: (more info (https://github.com/docker-library/official-images#architectures-other-than-amd64))
  `amd64` (https://hub.docker.com/r/amd64/mongo/), `arm64v8` (https://hub.docker.com/r/arm64v8/mongo/), `windows-amd64` (https://hub.docker.com/r/winamd64/mongo/)

- **Published image artifact details**:
  repo-info repo's `repos/mongo/` directory (https://github.com/docker-library/repo-info/blob/master/repos/mongo) (history (https://github.com/docker-library/repo-info/commits/master/repos/mongo))
  (image metadata, transfer size, etc)

- **Image updates**:
  official-images PRs with label `library/mongo` (https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fmongo)
  official-images repo's `library/mongo` file (https://github.com/docker-library/official-images/blob/master/library/mongo) (history (https://github.com/docker-library/official-images/commits/master/library/mongo))

- **Source of this description**:
  docs repo's `mongo/` directory (https://github.com/docker-library/docs/tree/master/mongo) (history (https://github.com/docker-library/docs/commits/master/mongo))

- **Supported Docker versions**:
  the latest release (https://github.com/docker/docker-ce/releases/latest) (down to 1.6 on a best-effort basis)

What is MongoDB?

MongoDB is a [free and open-source cross-platform document-oriented database (https://en.wikipedia.org/wiki/Document-oriented_database)](https://en.wikipedia.org/wiki/Document-oriented_database) program. Classified as a [NoSQL (https://en.wikipedia.org/wiki/NoSQL)](https://en.wikipedia.org/wiki/NoSQL) database program, MongoDB uses (JSON) [https://en.wikipedia.org/wiki/JSON]-like (https://en.wikipedia.org/wiki/JSON]-like) documents with [schemata (https://en.wikipedia.org/wiki/Database_schema)](https://en.wikipedia.org/wiki/Database_schema). MongoDB is developed by [MongoDB Inc. (https://en.wikipedia.org/wiki/MongoDB_Inc.)](https://en.wikipedia.org/wiki/MongoDB_Inc.), and is published under a combination of the [Server Side Public License (https://www.mongodb.com/licensing/server-side-public-license)](https://www.mongodb.com/licensing/server-side-public-license) and the [Apache License (https://en.wikipedia.org/wiki/Apache_License)](https://en.wikipedia.org/wiki/Apache_License).

First developed by the software company 10gen (now MongoDB Inc.) in October 2007 as a component of a planned platform as a service product, the company shifted to an open source development model in 2009, with 10gen offering commercial support and other services. Since then, MongoDB has been adopted as backend software by a number of major websites and services, including MetLife, Barclays, ADP, UPS, Viacom, and the New York Times, among others. MongoDB is the most popular NoSQL database system.

> wikipedia.org/wiki/MongoDB (https://en.wikipedia.org/wiki/MongoDB)

How to use this image

Start a `mongo` server instance

```
$ docker run --name some-mongo -d mongo:tag
```

... where `some-mongo` is the name you want to assign to your container and tag is the tag specifying the MongoDB version you want. See the list above for relevant tags.

Connect to MongoDB from another Docker container

The MongoDB server in the image listens on the standard MongoDB port, `27017`, so connecting via container linking or Docker networks will be the same as connecting to a remote `mongod`. The following example starts another MongoDB container instance and runs the `mongo` command line client against the original MongoDB container from the example above, allowing you to execute MongoDB statements against your database instance:

```
$ docker run -it --link some-mongo:mongo --rm mongo mongo --host mongo test
```

... where `some-mongo` is the name of your original `mongo` container.

... via `docker stack deploy` (https://docs.docker.com/engine/reference/commandline/stack_deploy/) or `docker-compose` (https://github.com/docker/compose)

Example `stack.yml` for `mongo`:

```
# Use root/example as user/password credentials
version: '3.1'

services:

  mongo:
    image: mongo
    restart: always
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: example

  mongo-express:
    image: mongo-express
    restart: always
    ports:
      - 8081:8081
    environment:
      ME_CONFIG_MONGODB_ADMINUSERNAME: root
      ME_CONFIG_MONGODB_ADMINPASSWORD: example
```

Try in PWD (http://play-with-docker.com?stack=https://raw.githubusercontent.com/docker-library/docs/3a01591e2c903a8c3224fced78f3f22b817b6272/mongo/stack.yml)

Run docker stack deploy -c stack.yml mongo (or docker-compose -f stack.yml up), wait for it to initialize completely, and visit http://swarm-ip:8081 , http://localhost:8081 , or http://host-ip:8081 (as appropriate).

Container shell access and viewing MongoDB logs

The docker exec command allows you to run commands inside a Docker container. The following command line will give you a bash shell inside your mongo container:

```
$ docker exec -it some-mongo bash
```

The MongoDB Server log is available through Docker's container log:

```
$ docker logs some-mongo
```

Configuration

See the MongoDB manual (https://docs.mongodb.com/manual/) for information on using and configuring MongoDB for things like replica sets and sharding.

Customize configuration without configuration file

Most MongoDB configuration can be set through flags to mongod . The entrypoint of the image is created to pass its arguments along to mongod . See below an example of setting MongoDB to use a smaller default file size (https://docs.mongodb.com/manual/reference/program/mongod/#cmdoption-mongod-smallfiles) via docker run .

```
$ docker run --name some-mongo -d mongo --smallfiles
```

And here is the same with a docker-compose.yml file

```
version: '3.1'
services:
  mongo:
    image: mongo
    command: --smallfiles
```

To see the full list of possible options, check the MongoDB manual on mongod
(https://docs.mongodb.com/manual/reference/program/mongod/) or check the `--help` output of `mongod`:

```
$ docker run -it --rm mongo --help
```

Using a custom MongoDB configuration file

For a more complicated configuration setup, you can still use the MongoDB configuration file. `mongod` does not
read a configuration file by default, so the `--config` option with the path to the configuration file needs to be
specified. Create a custom configuration file and put it in the container by either creating a custom Dockerfile
`FROM mongo` or mounting it from the host machine to the container. See the MongoDB manual for a full list of
configuration file (https://docs.mongodb.com/manual/reference/configuration-options/) options.

For example, `/my/custom/mongod.conf` is the path to the custom configuration file. Then start the
MongoDB container like the following:

```
$ docker run --name some-mongo -v /my/custom:/etc/mongo -d mongo --config /etc/mon
```

Environment Variables

When you start the `mongo` image, you can adjust the initialization of the MongoDB instance by passing one or
more environment variables on the `docker run` command line. Do note that none of the variables below will
have any effect if you start the container with a data directory that already contains a database: any pre-existing
database will always be left untouched on container startup.

MONGO_INITDB_ROOT_USERNAME , MONGO_INITDB_ROOT_PASSWORD

These variables, used in conjunction, create a new user and set that user's password. This user is created in the
`admin` authentication database and given the role of `root`. Both variables are required for a user to be
created. If both are present then MongoDB will start with authentication enabled: `mongod --auth`.
Authentication in MongoDB is fairly complex, so more complex user setup is explicitly left to the user via
`/docker-entrypoint-initdb.d/` (see *Initializing a fresh instance* below). The following is an example of
using these two variables to create a MongoDB instance and then using the `mongo` cli to connect against the
`admin` authentication database.

```
$ docker run -d --name some-mongo -e MONGO_INITDB_ROOT_USERNAME=mongoadmin -e MONG

$ docker run -it --rm --link some-mongo:mongo mongo mongo --host mongo -u mongoadmi
> db.getName();
some-db
```

If you do not provide these two variables or do not set the `--auth` flag with your own custom user setup, then
MongoDB will not require authentication. For more details about the functionality described here, please see the
sections in the official documentation which describe authentication
(https://docs.mongodb.com/manual/core/authentication/) and authorization
(https://docs.mongodb.com/manual/core/authorization/) in more detail.

MONGO_INITDB_DATABASE

This variable allows you to specify the name of a database to be used for creation scripts in `/docker-entrypoint-initdb.d/*.js` (see *Initializing a fresh instance* below). MongoDB is fundamentally designed for "create on first use", so if you do not insert data with your JavaScript files, then no database is created.

Docker Secrets

As an alternative to passing sensitive information via environment variables, `_FILE` may be appended to the previously listed environment variables, causing the initialization script to load the values for those variables from files present in the container. In particular, this can be used to load passwords from Docker secrets stored in `/run/secrets/<secret_name>` files. For example:

```
$ docker run --name some-mongo -e MONGO_INITDB_ROOT_PASSWORD_FILE=/run/secrets/mon
```

Currently, this is only supported for `MONGO_INITDB_ROOT_USERNAME` and `MONGO_INITDB_ROOT_PASSWORD`.

Initializing a fresh instance

When a container is started for the first time it will execute files with extensions `.sh` and `.js` that are found in `/docker-entrypoint-initdb.d`. Files will be executed in alphabetical order. `.js` files will be executed by `mongo` using the database specified by the `MONGO_INITDB_DATABASE` variable, if it is present, or `test` otherwise. You may also switch databases within the `.js` script.

Caveats

Where to Store Data

Important note: There are several ways to store data used by applications that run in Docker containers. We encourage users of the `mongo` images to familiarize themselves with the options available, including:

- Let Docker manage the storage of your database data <u>by writing the database files to disk on the host system using its own internal volume management (https://docs.docker.com/engine/tutorials/dockervolumes/#adding-a-data-volume)</u>. This is the default and is easy and fairly transparent to the user. The downside is that the files may be hard to locate for tools and applications that run directly on the host system, i.e. outside containers.
- Create a data directory on the host system (outside the container) and <u>mount this to a directory visible from inside the container (https://docs.docker.com/engine/tutorials/dockervolumes/#mount-a-host-directory-as-a-data-volume)</u>. This places the database files in a known location on the host system, and makes it easy for tools and applications on the host system to access the files. The downside is that the user needs to make sure that the directory exists, and that e.g. directory permissions and other security mechanisms on the host system are set up correctly.

WARNING (Windows & OS X): The default Docker setup on Windows and OS X uses a VirtualBox VM to host the Docker daemon. Unfortunately, the mechanism VirtualBox uses to share folders between the host system and the Docker container is not compatible with the memory mapped files used by MongoDB (see <u>vbox bug (https://www.virtualbox.org/ticket/819)</u>, <u>docs.mongodb.org (https://docs.mongodb.com/manual/administration/production-notes/#fsync-on-directories)</u> and related <u>jira.mongodb.org (https://jira.mongodb.org/browse/SERVER-8600)</u> bug). This means that it is not possible to run a MongoDB container with the data directory mapped to the host.

The Docker documentation is a good starting point for understanding the different storage options and variations, and there are multiple blogs and forum postings that discuss and give advice in this area. We will simply show the basic procedure here for the latter option above:

1. Create a data directory on a suitable volume on your host system, e.g. `/my/own/datadir`.

2. Start your `mongo` container like this:

```
$ docker run --name some-mongo -v /my/own/datadir:/data/db -d mongo
```

The `-v /my/own/datadir:/data/db` part of the command mounts the `/my/own/datadir` directory from the underlying host system as `/data/db` inside the container, where MongoDB by default will write its data files.

This image also defines a volume for `/data/configdb` for use with `--configsvr` (see docs.mongodb.com for more details) (https://docs.mongodb.com/v3.4/reference/program/mongod/#cmdoption-configsvr).

Creating database dumps

Most of the normal tools will work, although their usage might be a little convoluted in some cases to ensure they have access to the `mongod` server. A simple way to ensure

---

Docker Pull Command 📋

```
docker pull mongo
```