



ALLEGHENY
COLLEGE

Introduction
to Database
Systems:
CS305
Setting up
MongoDB
with Docker

Oliver
Bonham-
Carter
Hang Zhao

Install Docker

Start
MongoDB
MacOS and
Linux

Try out the
database!

Introduction to Database Systems: CS305

Setting up MongoDB with Docker

Oliver Bonham-Carter
Hang Zhao

16 November 2023

Setup Docker

Read the Docker Install Docs

Introduction
to Database
Systems:
CS305
Setting up
MongoDB
with Docker

Oliver
Bonham-
Carter
Hang Zhao

Install Docker

Start
MongoDB
MacOS and
Linux

Try out the
database!

Docker Desktop works with your choice of development tools and languages and gives you access to a vast library of certified images and templates in [Docker Hub](#). This allows development teams to extend their environment to rapidly auto-build, continuously integrate, and collaborate using a secure repository.



Install Docker Desktop

Install Docker Desktop on [Mac](#), [Windows](#), or [Linux](#).



Explore Docker Desktop

Navigate Docker Desktop and learn about its key features.



View the release notes

Find out about new features, improvements, and bug fixes.



Browse common FAQs

Explore general FAQs or FAQs for specific platforms.



Find additional resources

Find information on networking features, deploying on Kubernetes, and more.



Give feedback

Provide feedback on Docker Desktop or Docker Desktop features.

<https://docs.docker.com/desktop/>

Setup Mongo in Docker Container

Build the data directory

Introduction
to Database
Systems:
CS305
Setting up
MongoDB
with Docker

Oliver
Bonham-
Carter
Hang Zhao



MacOS and Linux: Create a directory for data to persist.

```
mkdir -p ~/mongodata
```

Windows: Create a directory for data to persist.

```
mkdir C:\Users\olive\OneDrive\mongodata
```

Replace *olive* with your own windows username

Install Docker

Start
MongoDB

MacOS and
Linux

Try out the
database!

Setup Mongo in Docker Container

Construct the container

Introduction
to Database
Systems:
CS305
Setting up
MongoDB
with Docker

Oliver
Bonham-
Carter
Hang Zhao

Install Docker

Start
MongoDB
MacOS and
Linux

Try out the
database!

MacOS and Linux: Start a Docker container by downloading from DockerHub

```
sudo docker run -it -v ~/mongodata:/data/db --name mongodb -d mongo
```

Windows: Start a Docker container by downloading from DockerHub

```
docker run -it -v C:\Users\olive\OneDrive\mongodata:/data/db  
--name mongodb -d mongo
```

For **Windows** commands (below), remove the term:

sudo

Setup Mongo in Docker Container

Maintenance commands

Introduction
to Database
Systems:
CS305
Setting up
MongoDB
with Docker

Oliver
Bonham-
Carter
Hang Zhao

Install Docker

Start
MongoDB
MacOS and
Linux

Try out the
database!

Check log to see that the server is operational

```
sudo docker logs mongodb
```

Check status: Is the mongodb running?

```
sudo docker ps
```

If mongodb container is not running, then start it up

```
sudo docker start mongodb
```

To stop the mongodb container at the end of your work

```
sudo docker stop mongodb
```

Run instance of MongoDB, goes into root of container.

```
sudo docker exec -it mongodb bash
```

Setup Mongo in Docker Container

From inside the container

Introduction
to Database
Systems:
CS305
Setting up
MongoDB
with Docker

Oliver
Bonham-
Carter
Hang Zhao

Install Docker

Start
MongoDB
MacOS and
Linux

Try out the
database!

You are now able to run MongoDB commands here.

Start the MongoDB client

```
mongosh
```

Leave the container

```
exit
```

Setup Mongo in Docker Container

Leaving/removing the container

Introduction
to Database
Systems:
CS305
Setting up
MongoDB
with Docker

Oliver
Bonham-
Carter
Hang Zhao

Install Docker

Start
MongoDB
MacOS and
Linux

Try out the
database!

Stop MongoDB container

```
sudo docker stop mongodb
```

Removing all stopped containers, if necessary due to errors in launching container

```
sudo docker rm $(docker ps -a -q)
```

Populate the MongoDB database

```
db.schools.insertMany([
  { 'school': "Washington", name: 'Ryan', gender: 'M', 'Job':'Teacher' },
  { 'school': "Edison", name: 'Joanna', gender: 'F', 'Job':'Professor'},
  { 'school': "Eaton", name: 'Roger', gender: 'M', 'Job':'Instructor'},
  { 'school': "Lewis", name: 'Presilla', gender: 'F', 'Job':'Instructor'}
]);
```

General Syntax

```
db.schools.find({ SEARCH-SPACE }, {CONSTRAINTS} )
```

- SEARCH-SPACE → Scan this search space
- CONSTRAINTS → Find constraint(s) within the search space

Querying “Jobs” == “Instructor”, show *name*, *gender* and *school*

```
db.schools.find(['Job':'Instructor'],
  {'name': 1, 'gender': 1, 'school':1}).pretty()
```