# Deploying an App to Heroku (webserver) and MongoDB Atlas (database)

Want to transfer our database from mongodb://localhost:27017 to and MongoDB Atlas web-database

AND to transfer the app built in section 28 from http://localhost:3000 to a Heroku webserver

## Transfer Database to MongoDB Atlas

1. Go to <https://mongodb.com>
2. Sign up for a MongoDB Atlas account (free tier): <https://www.mongodb.com/cloud/atlas/register1> (I used Google signup)
3. Create a free Cloud database: use AWS (recommended region – not paid), Cluster tier (free)
4. Create username and password (need to keep handy so app can connect to database)
   1. I used: user: admin-michael, password: ILovePasswords
   2. Under user privileges change to ‘Atlas Admin’
5. Connect from ‘Cloud Environment’
   1. Add IP address: 0.0.0.0/0
   2. This allows you to connect to the database from anywhere
6. Click Finish and close
7. In Database Deployments
   1. Click on the ‘Connect’ button to see how to connect to the Cluster
   2. Use ‘Connect with the MongoDB Shell’ option
   3. Click “I have the MongoDB Shell Installed” (it should be installed, if you did Section 28)
   4. In the terminal see what version you have by typing: > mongo –version
   5. Select the version from the dropdown list (select the one closest to your version)
   6. Copy the string over to the terminal to connect to the remote db
      1. Mine is:
      2. mongo "mongodb+srv://cluster0.3ppk9.mongodb.net/myFirstDatabase" --username admin-michael
      3. Copy the TEXT into the terminal to test it out (don’t use the button since it puts a ‘~’ at the end and messes it up)
      4. Use the password for the user
      5. You can see the databases and collections on the server (show dbs, use <db name>, show collections)
   7. In the website, click on ‘Collections’ then click ‘Add my own Data’
      1. Enter a database name and collection name
      2. I used: toDoDB, todos
      3. Go back to the terminal where you are logged in and type > show dbs
      4. You should see the new database and collection
      5. If you navigate to the database > use <db name> you can create a new collection via > db.createCollection(“<collection name>”)
   8. Click on the ‘Database’ > ‘Connect’ button again and this time click on “Connect your application”
      1. My text is: mongodb+srv://admin-michael:<password>@cluster0.3ppk9.mongodb.net/myFirstDatabase?retryWrites=true&w=majority
      2. Copy the text to app.js and change the <PASSWORD> to the password you created
      3. Replace all the text that starts with ‘myFirstDatabase’ to the name of the database you created
      4. Mine looks like: michael:ILovePasswords@cluster0.3ppk9.mongodb.net/toDoDB

## Transfer Webserver to Heroku

1. Go to <https://heroku.com>
2. Sign-up / sign into Heroku account
3. Click on Documentation > NodeJS > Get started with NodeJS > I’m Ready to Start
4. Install the Heroku CLI
5. Click on ‘I have installed the Heroku CLI’
6. For an existing app, follow these instructions: <https://devcenter.heroku.com/articles/preparing-a-codebase-for-heroku-deployment>
   1. Prepare a git repo
      1. > git init
      2. > git add .
      3. > git commit -m “first commit”
   2. Add a Heroku remote repo
      1. > heroku login
      2. Enter credentials to log in (browser login)
      3. > heroku create (Creates remote repo called ‘heroku’ (origin) we push to)
      4. > touch Procfile (creates a Procfile)
      5. > echo “web: node app.js” > Procfile (writes the text to the file)
   3. Make the following change to the app.listen() function in app.js

let port = process.env.PORT; // Heroku port variable (works in Heroku server)

if (port == null || port == “”) {

Port = 3000;

}

* + 1. app.listen(**port** , function () { … });
    2. In the local app we usually use port 3000, however, Heroku uses different ports and process.env.PORT can be used to detect the port used by Heroku)
  1. Edit the package.json file to add the version of node used
     1. > node --version
     2. Add this text below “license”: “ISC”

“engines”: { “node”: “ <your version goes here>” },

* 1. Create a .gitignore file
     1. > touch .gitignore
     2. > nano .gitignore
     3. Add the following text

/node\_modules

Npm-debug.log

.DS\_Store

/\*.env

* + 1. Hit ctrl-x to save and exit
  1. Update the git repo
     1. > git add .
     2. > git commit -m “Add the .gitignore and Procfile, update PORT”
  2. Push the changes to the remote ‘heroku’ repo
     1. > git push Heroku master
     2. This will give you a URL of the new app/website

My to-do list app is at:

<https://ancient-chamber-55871.herokuapp.com/>

Can make custom to do lists by adding anything to the end of the name

E.g.,

<https://ancient-chamber-55871.herokuapp.com/work>

I have another one at:

<https://secure-earth-86865.herokuapp.com/>

This one connects to Mailchimp to keep a database of email address contacts