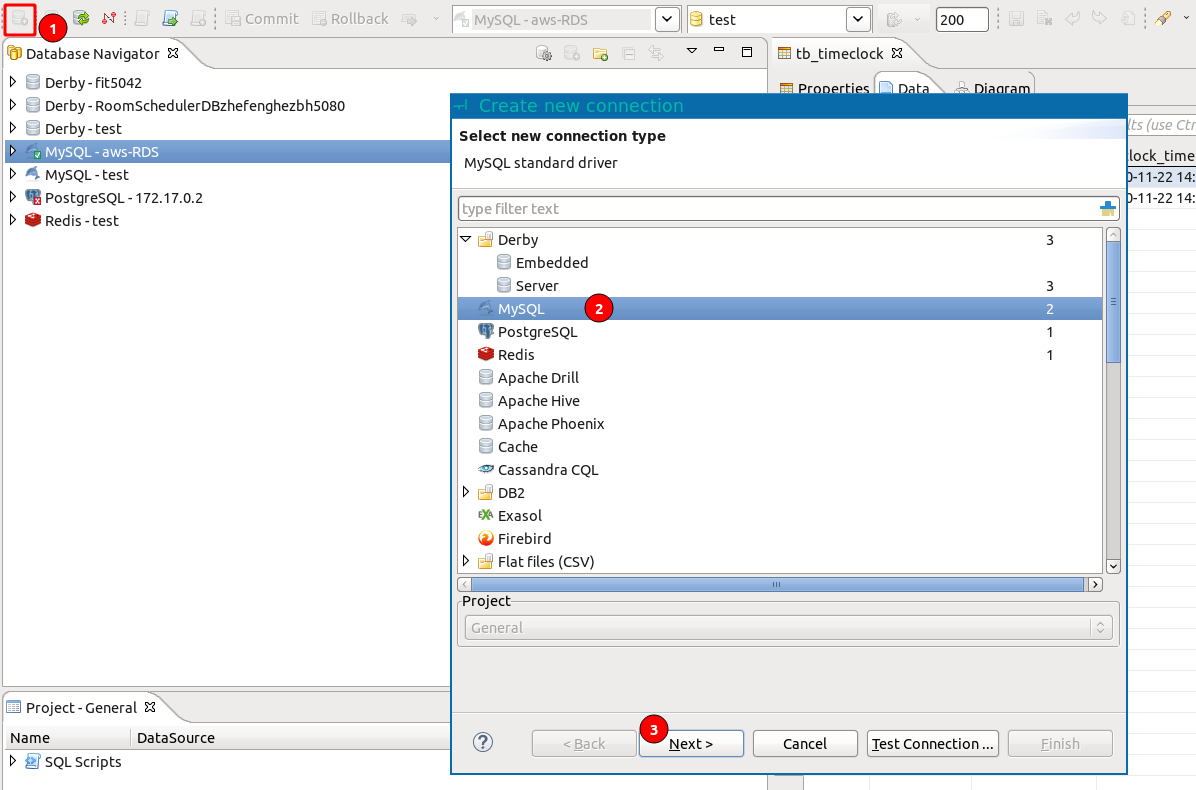
### Access the Web

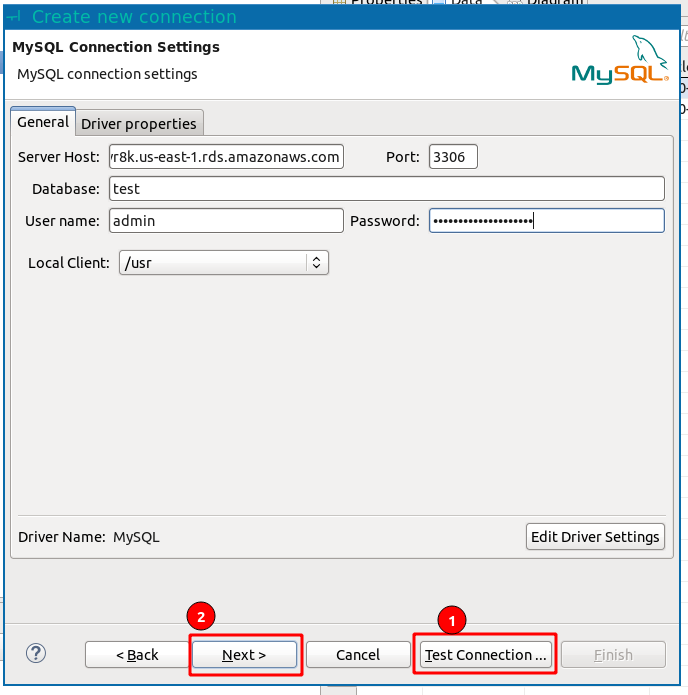
<http://3.238.131.94/>

(deployed on ec2 instance with docker, and port is 80)

### Check DB Data

任何mysql 客户端都可以(例如：[DBeaver](https://dbeaver.io/download/), 下载community版本就可以了).



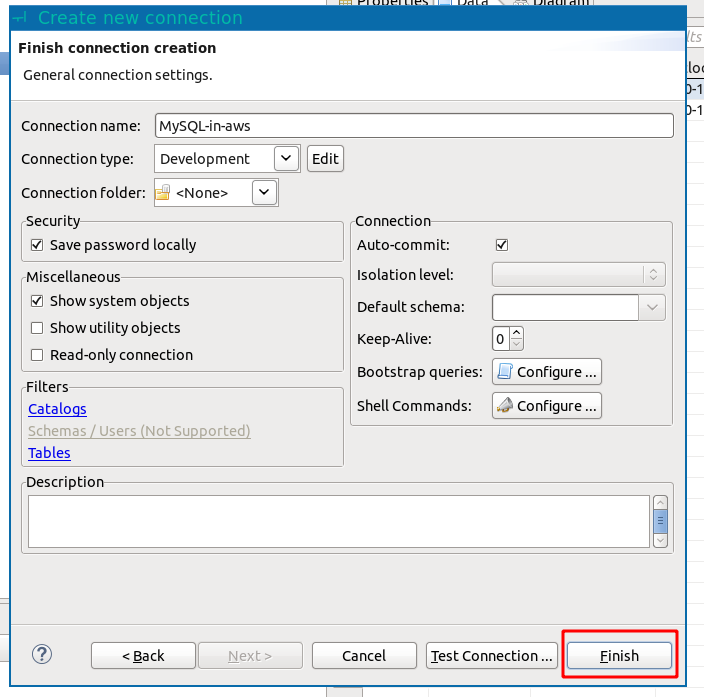


**host**: database-1.cwqvm1719ggg.us-east-1.rds.amazonaws.com

**username**: admin

**password**: dGT6!34SUL9!

**database**: booklib



### AWS Status

1. SSH登录AWS EC2:

ssh -i ec2-booklib.pem ubuntu@3.238.131.94

1. 查看docker运行状态：

docker ps

### Code

目录结构：

|  |
| --- |
| ├── backend  │   ├── config  │   │   └── config.json 本地开发，保存mysql连接信息  │   ├── entrypoint.sh build docker image时用到，docker的启动的入口  │   ├── index.js 后端最主要的js文件  │   ├── package.json nodejs项目meta文件，定义依赖等信息  │   └── static 此目录后面部署项目时，用来存放静态html以及资源  ├── frontend  │   ├── angular.json  │   ├── e2e  │   │   ├── protractor.conf.js  │   │   ├── src  │   │   │   ├── app.e2e-spec.ts  │   │   │   └── app.po.ts  │   │   └── tsconfig.json  │   ├── karma.conf.js  │   ├── package.json  │   ├── README.md  │   ├── src  │   │   ├── app  │   │   │   ├── app.component.css  │   │   │   ├── app.component.html  │   │   │   ├── app.component.ts  │   │   │   ├── app.module.ts  │   │   │   ├── app-routing.module.ts  │   │   │   ├── book-list  │   │   │   │   ├── book-list.component.css  │   │   │   │   ├── book-list.component.html 前端显示图书列表的组件html  │   │   │   │   └── book-list.component.ts 加载图书列表的js文件  │   │   │   ├── book.service.ts 调用后端API获取图书列表  │   │   │   └── book.ts 图书实体类  │   │   ├── assets  │   │   │   ├── bootstrap.min.css  │   │   │   ├── favicon.png  │   │   │   ├── images  │   │   │   │   ├── aws\_guide.jpg  │   │   │   │   ├── ...  │   │   │   └── library.jpg  │   │   ├── environments  │   │   │   ├── environment.prod.ts 定义apiUrl等环境变量  │   │   │   └── environment.ts  │   │   ├── index.html 整个前端最顶层的html文件  │   │   ├── main.ts  │   │   ├── polyfills.ts  │   │   └── styles.css  │   ├── tsconfig.app.json  │   ├── tsconfig.json  │   ├── tsconfig.spec.json  │   └── tslint.json  ├── README.md  └── scripts  ├── build-docker.sh <-------------------------创建docker image的脚本  ├── db.sql <--------------------------------创建数据库table的语句  ├── deploy <--------------------------------用来在aws上部署的目录  │   ├── config  │   │   └── config.json  │   └── deploy.sh <-------------------------在aws部署的脚本  └── docker  └── Dockerfile <-------------------------docker image定义文件 |

1. 前端：AngularJS工程

上面标红的文件比较重要，其他文件基本都是[angular-cli](https://cli.angular.io/)生成的框架代码。

angular-cli是为了方便创建和管理angular工程的命令行工具。

1. 安装angular-cli: npm install -g @angular/cli
2. 创建angular工程：ng new my-app
3. 生成部署所需的静态文件：ng build --prod
4. 后端：nodejs + express

[express](https://expressjs.com/)是nodejs的web框架，安装：npm install express --save

核心代码就在backend/index.js文件

1. 数据库

MySQL，创建表的SQL语句，查看scripts/db.sql

1. docker相关

查看image (scripts/build-docker.sh会编译生成image): docker images

查看docker运行的容器(container, deploy.sh会生成它)： docker ps

### Demo + Presentation

1. 展示和介绍UI

<http://3.238.131.94/>

1. 用客户端查看mysql数据表
2. 查看AWS EC2 和 RDS的状态

- 说明public ip以及DB连接信息

- SSH登录ec2，查看docker容器的状态 (docker ps)

ssh -i ec2-booklib.pem ubuntu@3.238.131.94

docker ps