The Terminal Deployment Documentation

Database Deployment

Launching a Database Instance

To setup the AWS Relational Database service hosted MySQL database. Follow the steps in the following tutorial:
http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_SettingUp.html

When arrived at the section titled "Determine Requirements" we will use a Default VPC with a user created security group. Follow the steps under "Provide Access to the DB Instance in the VPC by Creating a Security Group", choosing any values for Name tag, Group name, and description. When editing the Inbound Rules, choose type "MySQL/Aurora", protocol "TCP (6)". Port Range should be auto completed as "3306", if it is not type in "3306" as the port range, for Source use address "0.0.0.0/0".

We will not be adding any additional Inbound or Outbound rules for this security group.

Once the security group has been created, follow the steps in the following tutorial to setup the MySQL Database instance:
http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.CreatingConnecting.MySQL.html

After selecting "Launch DB Instance" and "MySQL' as the database engine, check the box "Only enable options eligible for RDS Free Usage Tier" to only allow free options to be selected. If this box is not checked, on the "Use case" page choose Dev/Test - MySQL.

When on the "Specify DB Details" page, use the following details:
- For DB Instance class choose any DB instance desired. Please be aware that larger instances will cost more. For the purpose of this project we chose "db.t2.micro".
- For Multi-AZ Deployment, choose "No".
- For Storage type, choose "General Purpose (SSD)"
- For allocated storage, input 20GB.
- For DB Instance Identifier, choose a unique name. For the purpose of this project we chose "the-terminal-db-instance".
- For Master Username, choose any username and make note of it. For the purpose of this project we chose "Administrator".
- For Master password, choose any password and make note of it. For the purpose of this project we chose the password "TheTerminal!".

For every other field under "Specify DB Details" page, choose the values given in the tutorial.

When on the "Configure advanced settings" page, use the following details:
- For VPC, choose "Default VPC"

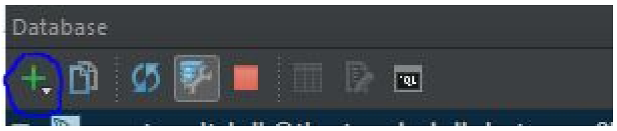
- For Subnet group, choose “default”
- For public accessibility, choose “Yes”
- For availability zone, choose “No preference”
- For VPC Security groups, choose “Select existing VPC security groups” and choose the security group that was previously created in the “Provide Access to the DB Instance in the VPC by Creating a Security Group” step. The newly created security group along with “default (VPC)” should be shown underneath the “Select VPC Security groups” dropdown.
- For Database name, choose any value for database name. For the purpose of this project we used “coast_capital_db”
- For Database port, keep the default port of 3306.
- For “IAM DB authentication”, select disable.
- If “Only enable options eligible for RDS Free Usage Tier” was selected previously, encryption cannot be enabled so leave that as default (disabled).
- For Backup, choose a backup retention period of 7 days and no preference for backup window.
- For Monitoring, choose “Disable enhanced monitoring”
- For Maintenance, choose “Enable auto minor version upgrade” and select no preference for “Maintenance window”.

Select “Launch DB Instance”. A DB instance will now be launched (note this may take a few minutes). Wait for the status to change to “available” before attempting to connect to the database instance.

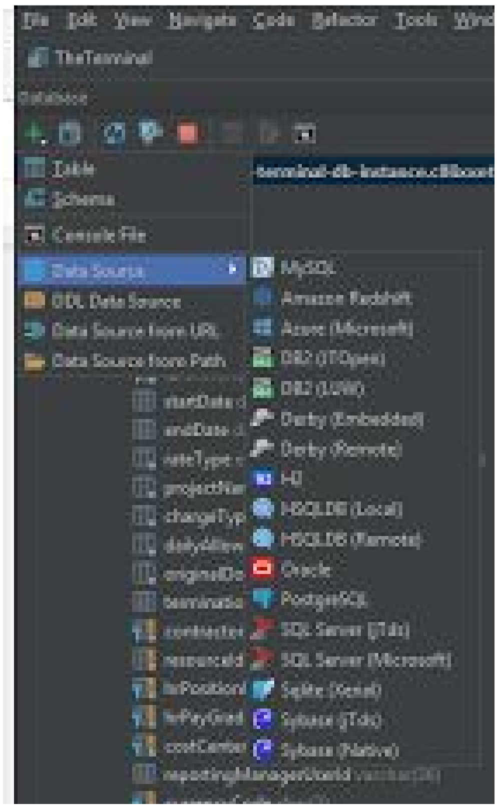
Connecting to the newly launched database instance

For the purpose of this project, we used DataGrip to connect to our database instance. DataGrip can be installed through the following link:
https://www.jetbrains.com/datagrip/.

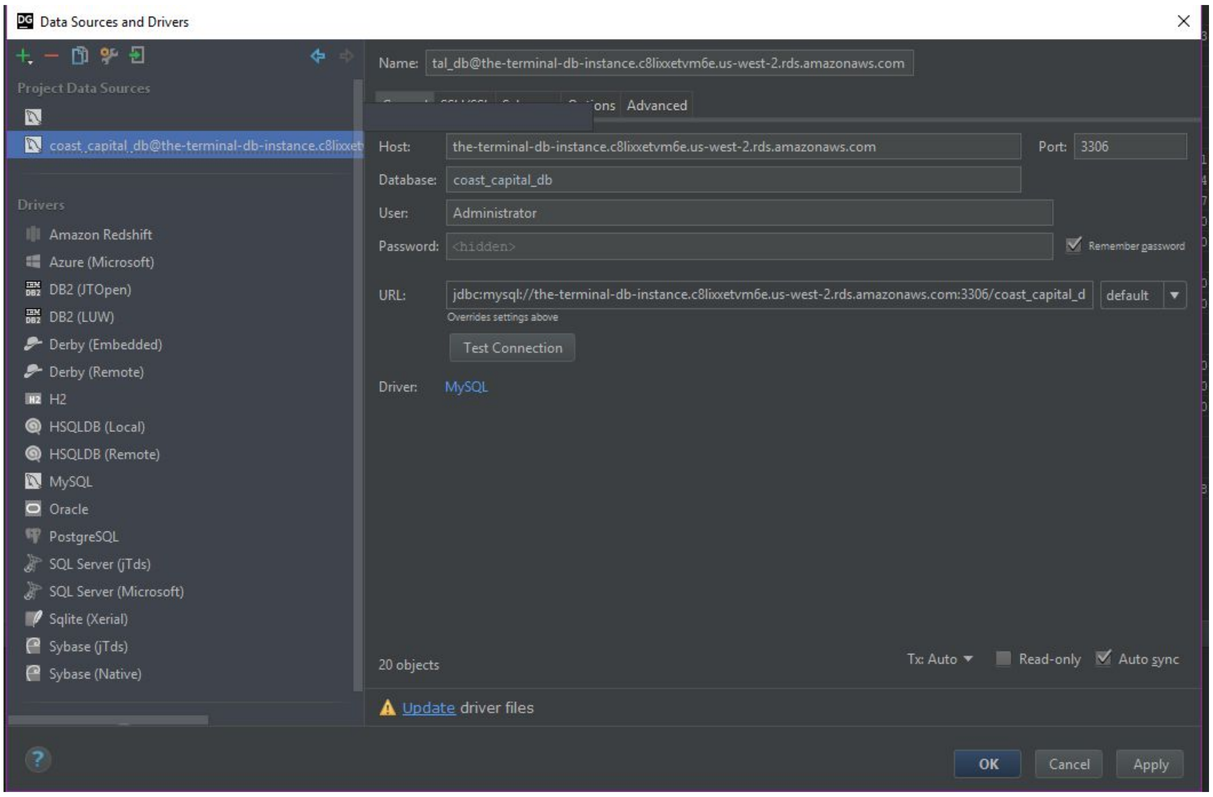
Once DataGrip has been installed it can be used to connect to the newly created database instance. To connect to the instance in DataGrip, under Database click the green “+” symbol, shown here:



Navigate to “Data Source” and choose “MySQL” as shown below:



Enter in the values for Host, Database, User and Password that were specified in the “Launching a Database Instance” step. For the purposes of this project, our values were as follows:



Name: tal_db@the-terminal-db-instance.c8lixxetvm6e.us-west-2.rds.amazonaws.com

Host: the-terminal-db-instance.c8lixxetvm6e.us-west-2.rds.amazonaws.com Port: 3306

Database: coast_capital_db

User: Administrator

Password: <hidden> ✓ Remember password

URL: jdbc:mysql://the-terminal-db-instance.c8lixxetvm6e.us-west-2.rds.amazonaws.com:3306/coast_capital_d default

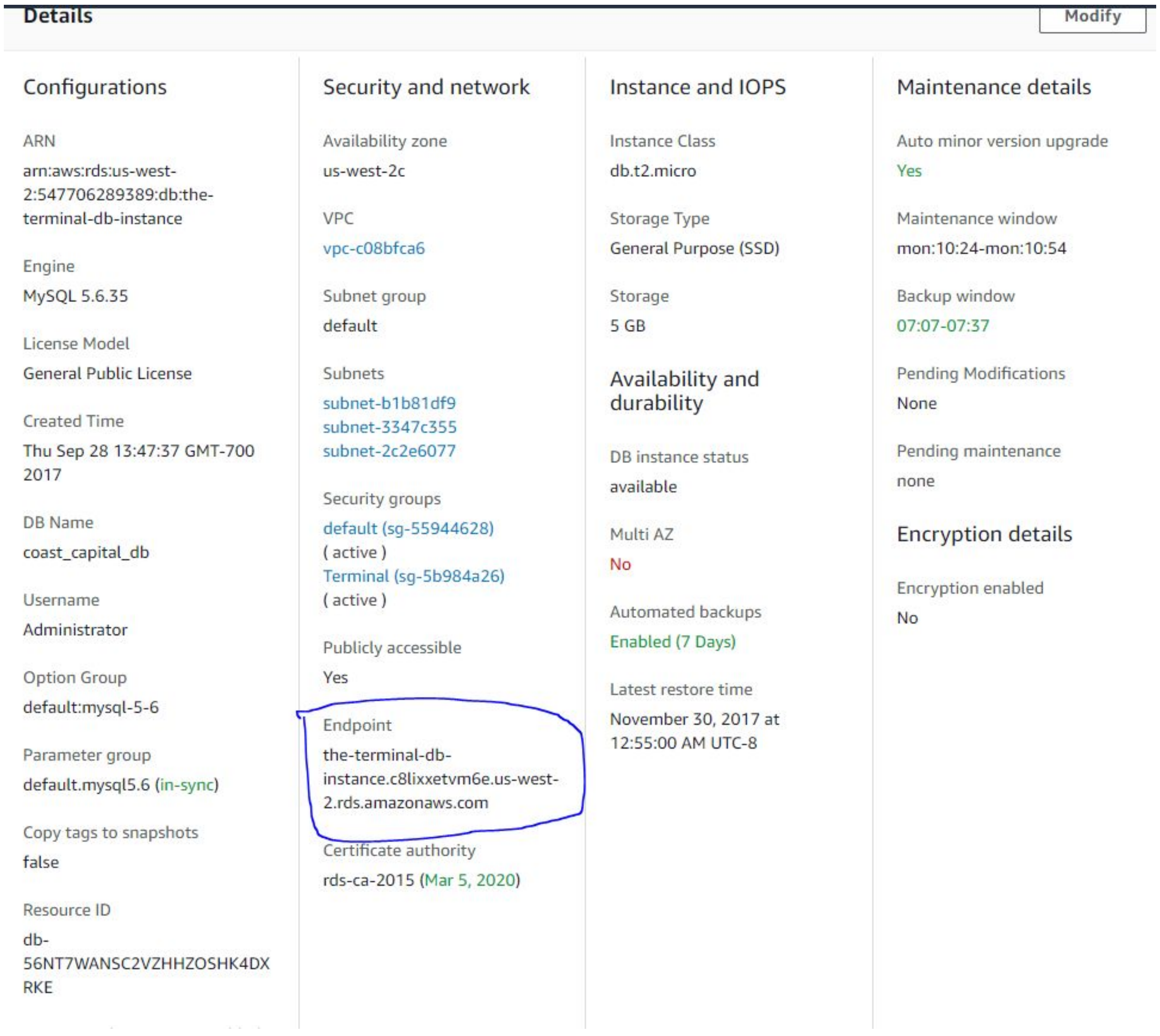
Overrides settings above

Driver: MySQL

20 objects

Tx: Auto Read-only ✓ Auto sync

The “Host” field value can be found on the AWS RDS management console. Select “Instances” and select the newly created database instance. The hostname is shown under the “Details” section, as shown below:


Details

Modify

Configurations

ARN
arn:aws:rds:us-west-2:547706289389:db:the-terminal-db-instance

Engine
MySQL 5.6.35

License Model
General Public License

Created Time
Thu Sep 28 13:47:37 GMT-700 2017

DB Name
coast_capital_db

Username
Administrator

Option Group
default:mysql-5-6

Parameter group
default.mysql5.6 (in-sync)

Copy tags to snapshots
false

Resource ID
db-56NT7WANSC2VZHHZOSHK4DXRKE

Security and network

Availability zone
us-west-2c

VPC
vpc-c08bfca6

Subnet group
default

Subnets
subnet-b1b81df9
subnet-3347c355
subnet-2c2e6077

Security groups
default (sg-55944628)
(active)
Terminal (sg-5b984a26)
(active)

Publicly accessible
Yes

Endpoint
the-terminal-db-instance.c8lixxetvm6e.us-west-2.rds.amazonaws.com

Certificate authority
rds-ca-2015 (Mar 5, 2020)

Instance and IOPS

Instance Class
db.t2.micro

Storage Type
General Purpose (SSD)

Storage
5 GB

Availability and durability

DB instance status
available

Multi AZ
No

Automated backups
Enabled (7 Days)

Latest restore time
November 30, 2017 at 12:55:00 AM UTC-8

Maintenance details

Auto minor version upgrade
Yes

Maintenance window
mon:10:24-mon:10:54

Backup window
07:07-07:37

Pending Modifications
None

Pending maintenance
none

Encryption details

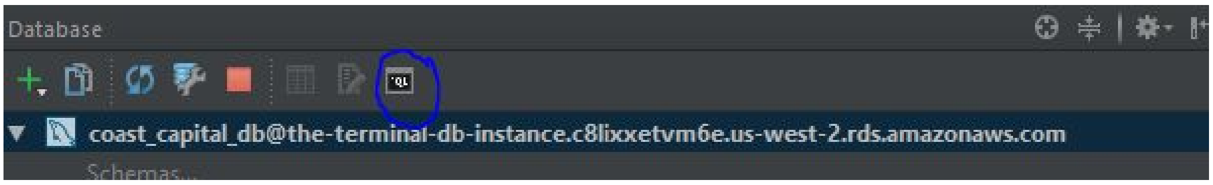
Encryption enabled
No

Input the values into their respective fields in the DataGrip add data source popup window. If a driver file is missing an error will be displayed at the bottom of the data source popup window, select “Download missing driver files” and DataGrip will automatically download the missing drivers required to connect to the database.

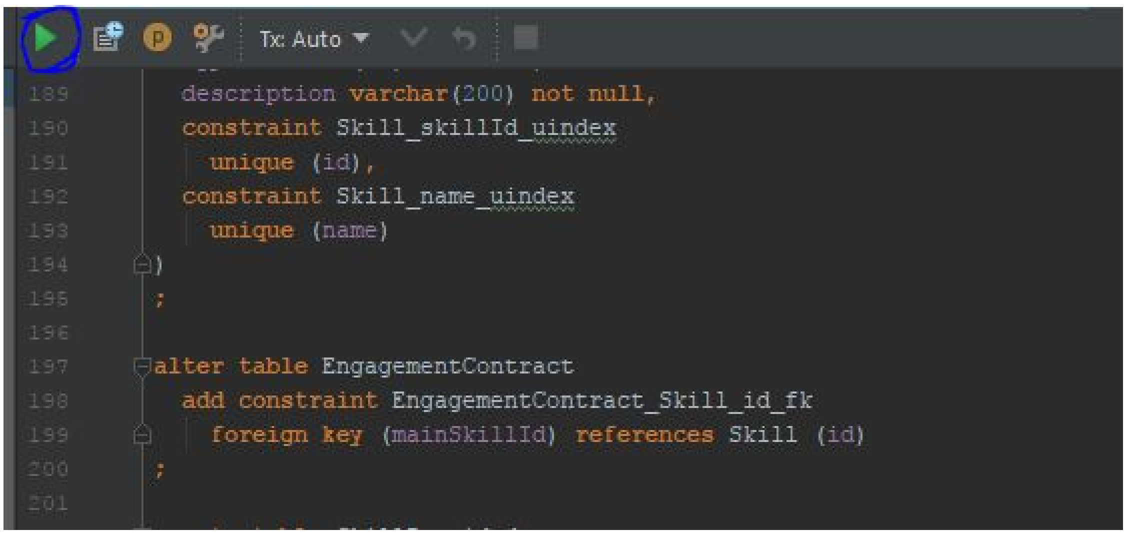
Creating the database tables

Once connected to the newly created database in DataGrip, creating the necessary database tables is as simple as running a SQL command.

Once connected to the database in Datagrip, click the terminal button to open a new SQL terminal, as shown below:


Database
coast_capital_db@the-terminal-db-instance.c8lixxetvm6e.us-west-2.rds.amazonaws.com
Schemas...

Copy and paste the commands from code/database/database.sql into the newly opened SQL terminal window. Select run (as shown below) to run the commands and create the database tables:



```
    description varchar(200) not null,
    constraint Skill_skillId_uindex
        unique (id),
    constraint Skill_name_uindex
        unique (name)
)
;

alter table EngagementContract
    add constraint EngagementContract_Skill_id_fk
        foreign key (mainSkillId) references Skill (id)
;
```

The necessary tables will now be created in the database.

Back-end Deployment

Generate the back-end deployable jar file

To generate the back-end project as a deployable jar file, install Maven and JDK 8 on your machine:

1. Maven can be downloaded from (select the zip version for windows): https://maven.apache.org/download.cgihttps://maven.apache.org/install.html
2. To install Maven, follow the following installation guide for your OS: https://maven.apache.org/install.html
3. To verify Maven has been installed, open command prompt or terminal and type “mvn -version”, if “mvn is not recognized as an internal or external command, operable program or batch file.” is the output a path variable may need to be set pointing to the Maven directory. To accomplish this the instructions under the link “Setting path variables” can be followed.
4. JDK 8 can be installed from (select the file for your operating system) http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

5. Verify JDK installation by opening a command prompt or terminal and typing “javac -version”, if javac is an unrecognized command follow the instructions to set the path variable.

Setting path variables: https://www.java.com/en/download/help/path.xml

Once Maven and JDK 8 have been installed, the back-end project can be compiled to a deployable jar file:

1. Navigate to the code/backend directory in command prompt or terminal (accomplished using cd {DIRECTORY_NAME}).
2. Once in the code/backend directory, type “mvn package”. This will package the backend project (and dependencies) into an executable jar file.
3. The jar file will be located under code/backend/target/gs-rest-service-0.1.0.jar. This file will later be used for deployment on the web server.

Creating the back-end server

1. To create the back-end server, navigate to the AWS console located at: https://console.aws.amazon.com/
2. In the “Find a service by name” textbox, type “Elastic Beanstalk” and select it.
3. Once in the Elastic Beanstalk console, choose “Create a New Application”. Choose an application name and description. For the purpose of this project the application name was “TheTerminal” and the description was “The Terminal RESTful API”.
4. Under the “New Environment” page, select “Create web server”
5. For “Environment Type” choose predefined configuration as “Java’ and Environment type as “Single Instance”
6. For “Application Version” choose “Upload your own” and select the jar file that was created in the “Generate the back-end deployable jar file” step.
7. Choose an environment name and URL, for this project we used “theterminal-backend-env”. Enter a description.
8. In “Additional Resources” check the box “Create this environment inside a VPC”
9. For “Configuration Details” choose the following values:
    - Instance type: t1.micro
    - EC2 key pair: [none selected]
    - Email address: your email address
    - System type: Enhanced
    - Root volume type: (Container default)
    - Root volume size: (unchecked)
10. No environment tags are necessary to specify
11. For VPC configuration choose the following values:
    - VPC: default selected value
    - Associate Public IP Address: checked
    - AZ: “us-west-2a” EC2 checked
    - VPC security group: select the security group that was created in the Database creation phase.

12. Keep the default instance profile and service role.
13. Confirm the values and select “Launch”

The back-end application will now begin deploying. Once it is deployed we need to configure environment variables to connect to the database. To do this:

1. Navigate to the newly created elastic beanstalk environment in the Elastic Beanstalk AWS console.
2. Click “Configuration” on the right hand side
3. Edit “Software Configuration”
4. Scroll down to “Environment Properties” and enter in the values:

Property Name	**Property Value**
RDS_DB_NAME	Database name defined in database setup step. For this project ours is “coast_capital_db”.
RDS_HOSTNAME	The database hostname shown in the RDS console. For this project ours is “the-terminal-db-instance.c8lixxetvm6e.us-west-2.rds.amazonaws.com”
RDS_USERNAME	The database master username specified before. For this project ours is “Administrator”
RDS_PORT	3306
RDS_PASSWORD	The database master password specified before. For this project ours is “TheTerminal!”
SERVER_PORT	5000

5. Click “Apply” and Elastic Beanstalk will apply the new configuration settings to the server.

The back-end server is now deployed.

Front-end Deployment

Deploying the front-end is a much simpler process.

Generate the front-end deployable distribution folder

To compile the front-end and dependencies npm (node package manager) is required. Follow the instructions in the following link to install npm: https://www.npmjs.com/get-npm?utm_source=house&utm_medium=homepage&utm_campaign=free%20orgs&utm_term=Install%20npm

Once npm is installed, a front-end distribution can be created.

1. Navigate to the code/frontend folder
2. Type “npm i” to install required dependencies
3. Type “npm run build” to build the project for deployment
    - This will create a “dist” folder under code/frontend which will contain everything necessary for deployment

Deploy the front-end

1. Navigate to the AWS management console located at: https://console.aws.amazon.com/
2. Type in “S3” into the “Find a service by name” field and select “S3”
3. Select “Create Bucket”
    a. Enter a bucket name, for this project we used “theterminal”
    b. Choose a region, default is fine. Click next.
    c. Click next on “Set properties” page
    d. Click next on “Set permissions” page
    e. Click “Create bucket”
4. Click the newly created bucket and navigate to the Permissions page
5. Click “Bucket Policy”
6. Enter the following bucket policy (with the bucket name as BUCKET_NAME):

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AddPerm",
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::{BUCKET_NAME}/*"
      }
   ]
}
```

7. Click “Save”
8. Go to the Properties page and scroll down to “Static website hosting” and select it.

9. Click “Use this bucket to host a website” and type “index.html” for the field “Index document”, Click “Save”.
10. Navigate back to the S3 console and click on the newly created bucket
11. Click “Create Folder” and enter the name “images”, select “none” for encryption settings and click “save”
12. Click “Upload”, navigate to code/frontend/dist folder created in the “Generate the front-end deployable distribution folder” step and upload “index.html” and “bundle.js”
13. Click the newly created images folder (in S3) and select Upload
14. Upload all of the images in code/frontend/dist/images folder
15. The newly hosted website will be available at {BUCKET_NAME}.s3-website.{BUCKET_REGION}.amazonaws.com, where BUCKET_NAME is the name of the newly created bucket and BUCKET_REGION is the region which was selected in step 3.

The application is now deployed!