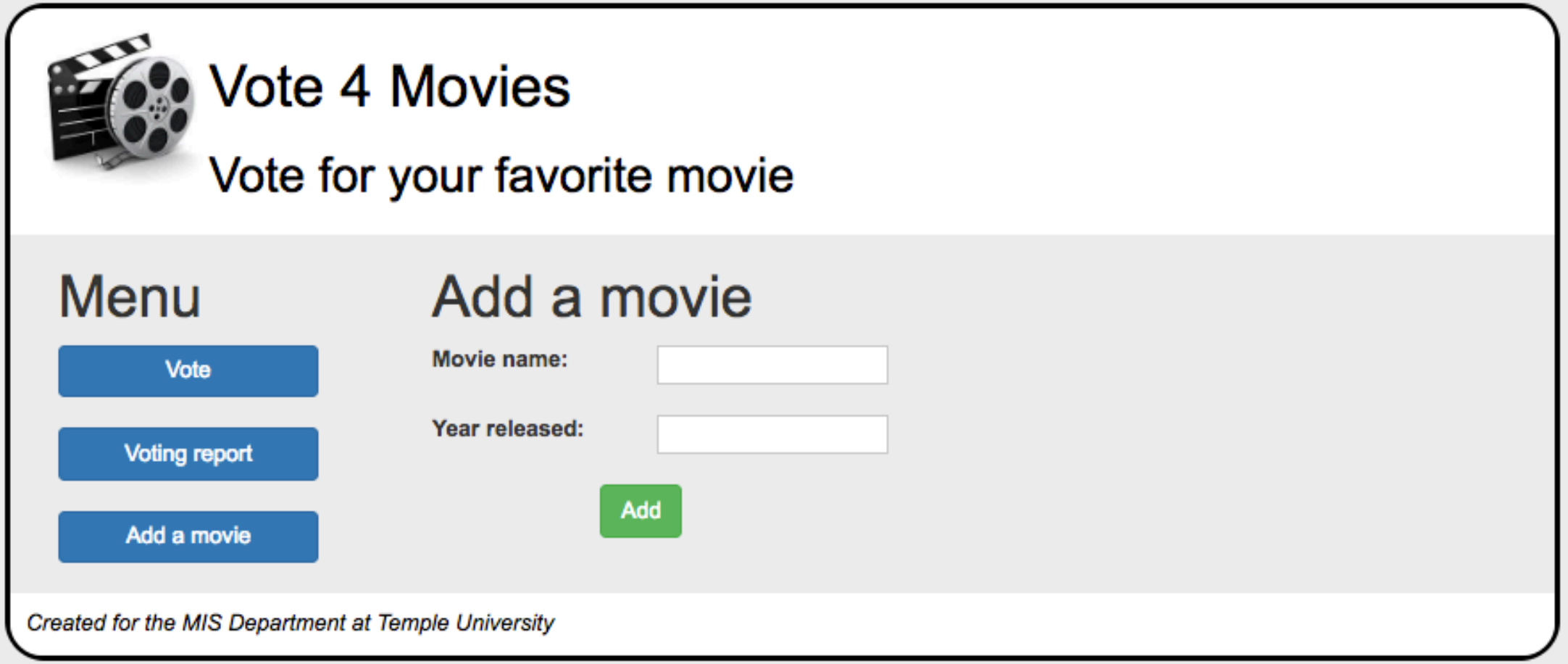
Creating Serverless Microservices on AWS

- Objective: Reproduce a (very simple) monolithic MVC application found at: https://misdemo.temple.edu/vote4movies

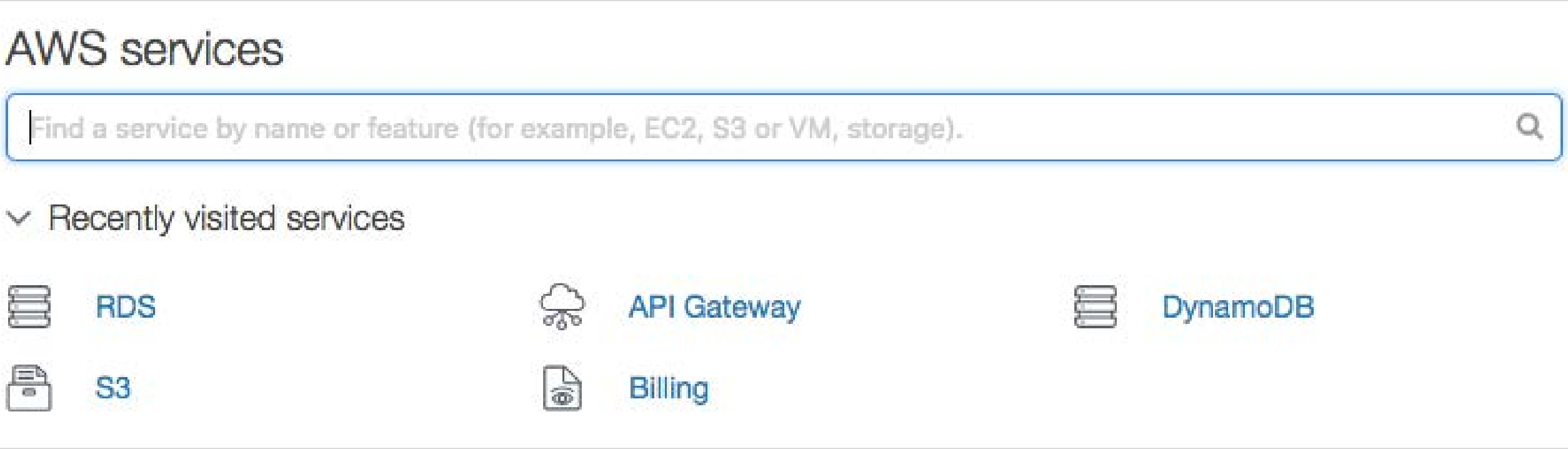
- Steps:
  - Define the database tier
  - Define the AWS Lambda functions
  - Define the Amazon API Gateway
  - Develop the application itself (HTML/CSS/JavaScript)
  - Post to Amazon S3

The original site



Define the database tier

Setting up my relational database …



Database options

Database name

shafervote4moviesdb

If you do not specify a database name, Amazon RDS does not create a database.

Database port

TCP/IP port the DB instance will use for application connections.

3306

DB parameter group Info

default.mariadb10.1

Option group Info

default:mariadb-10-1

Configurations

ARN

arn:aws:rds:us-east-2:646163223753:db:shaferrds

Engine

MariaDB 10.1.31

License Model

General Public License

DB Name

shafervote4moviesdb

Username

shaferrdsadmin

Option Group

default:mariadb-10-1

Parameter group

default.mariadb10.1 (in-sync)

Resource ID

db-KCUQOITR667HNCU7DTWP6MCINY

Security and network

Availability zone

us-east-2b

VPC

vpc-64a5e60c

Subnet group

default

Subnets

subnet-69621f01
subnet-6d4a5420
subnet-96d57fec

Security groups

rds-launch-wizard (sg-044638dce1a1dff3e)
(active)

Publicly accessible

No

Certificate authority

rds-ca-2015 (Mar 5, 2020)

Instance and IOPS

Instance Class

db.t2.micro

Storage Type

General Purpose (SSD)

Storage

20 GiB

Availability and durability

DB instance status

creating

Multi AZ

No

Backup and Restore

Automated backups

Enabled (7 Days)

Backup window

08:40-09:10 UTC (GMT)

Copy tags to snapshots

Yes

Maintenance details

Auto minor version upgrade

Yes

Maintenance window

sat:06:34-sat:07:04 UTC (GMT)

Pending Modifications

Master User Password: ****

Pending maintenance

none

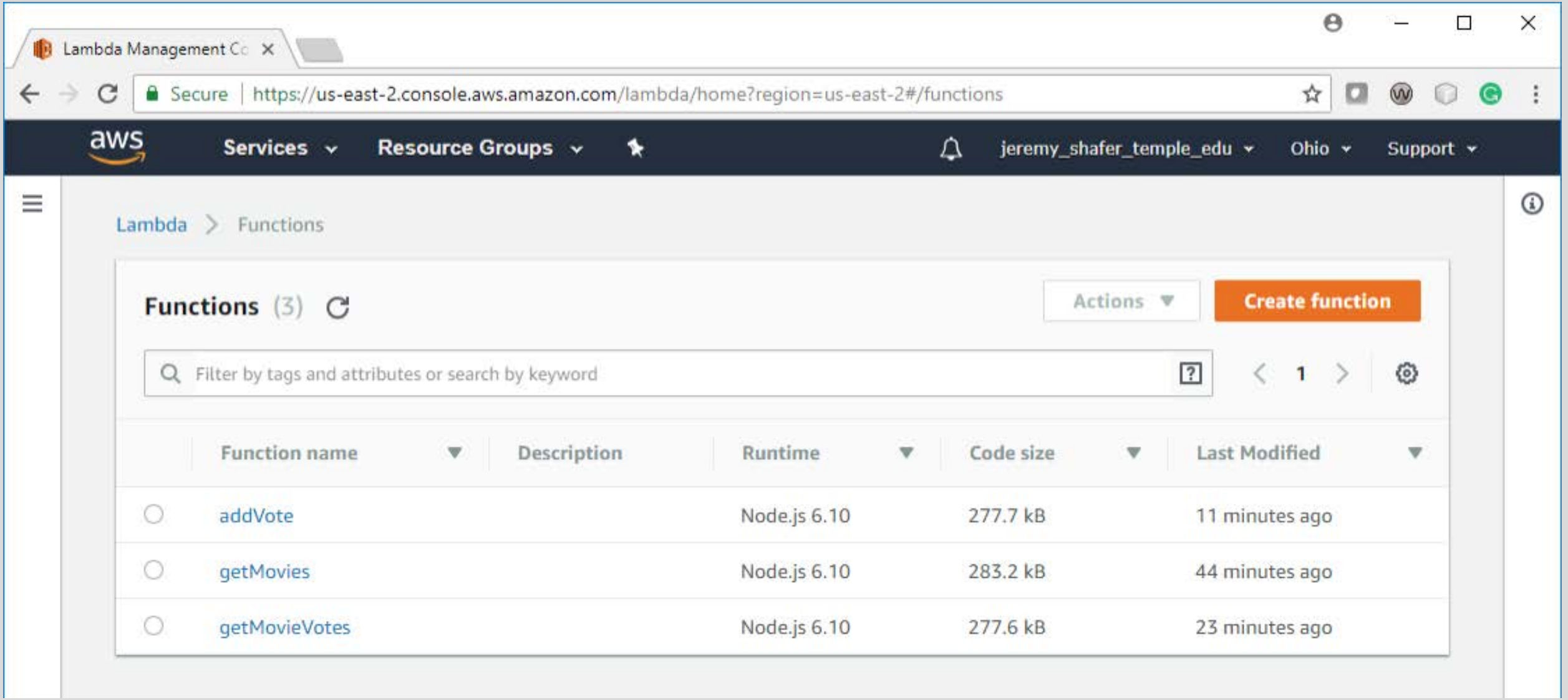
Encryption details

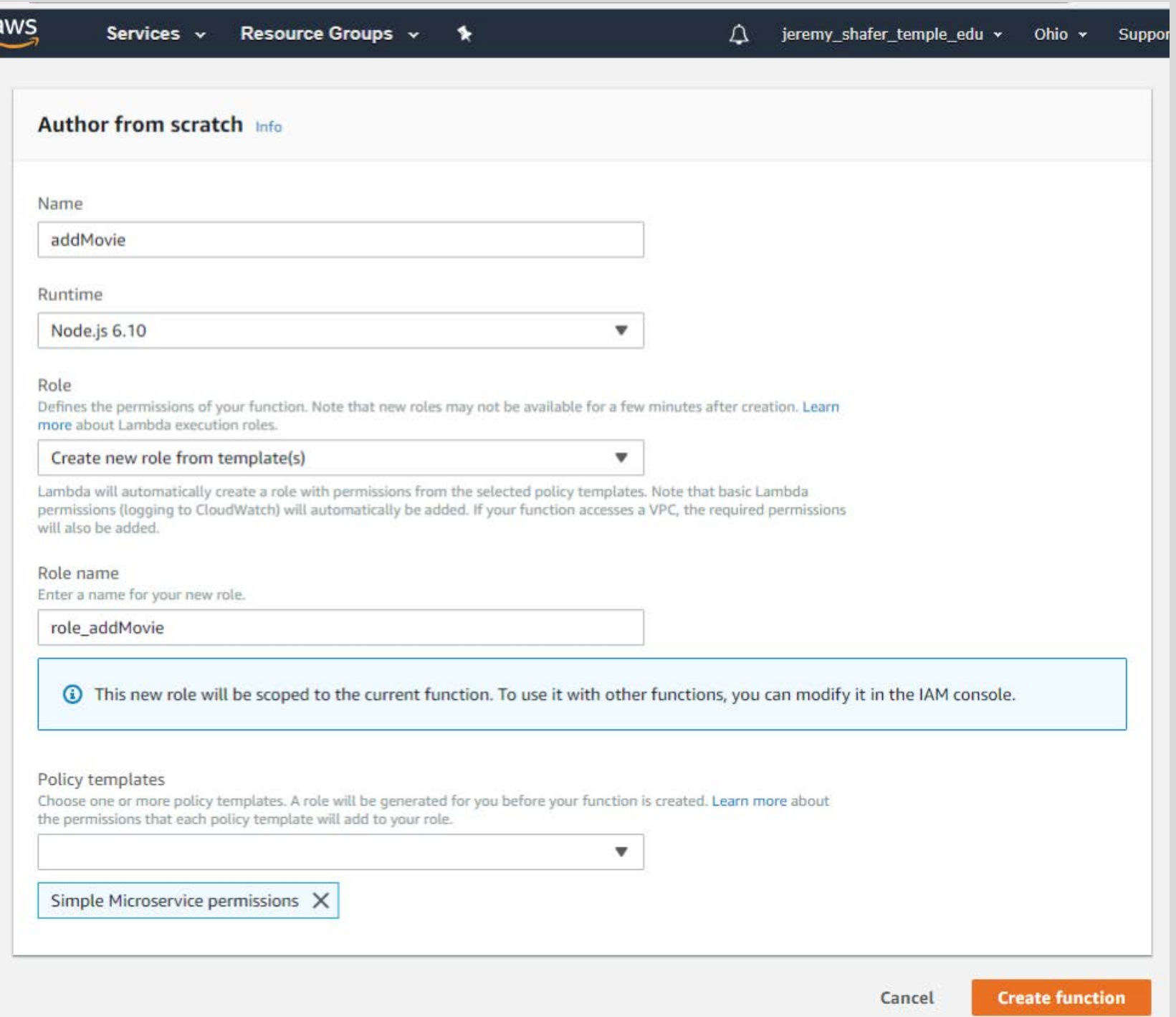
Encryption enabled

No

Define the AWS Lambda functions

Our simple application needed ***four*** functions.
So the following steps were repeated four times with appropriate variation.



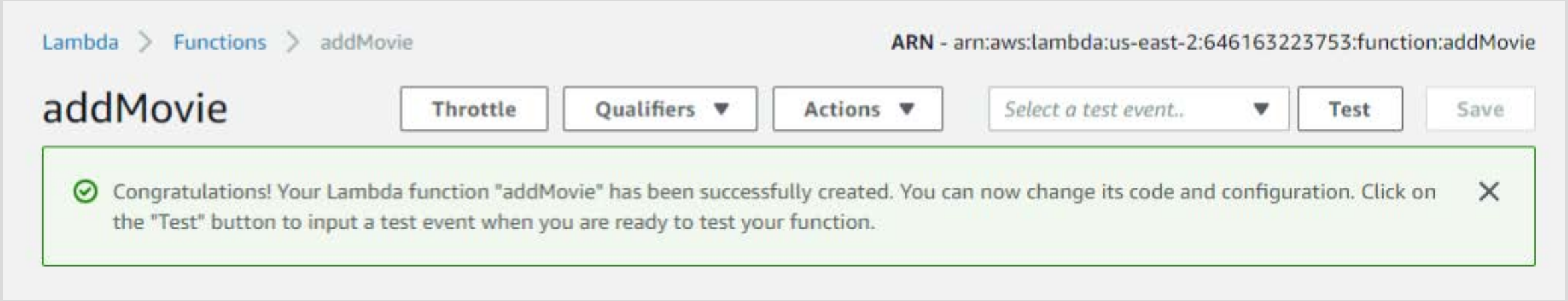


It was necessary to author the Lambda functions from scratch for a simple example that did not use some existing AWS service.

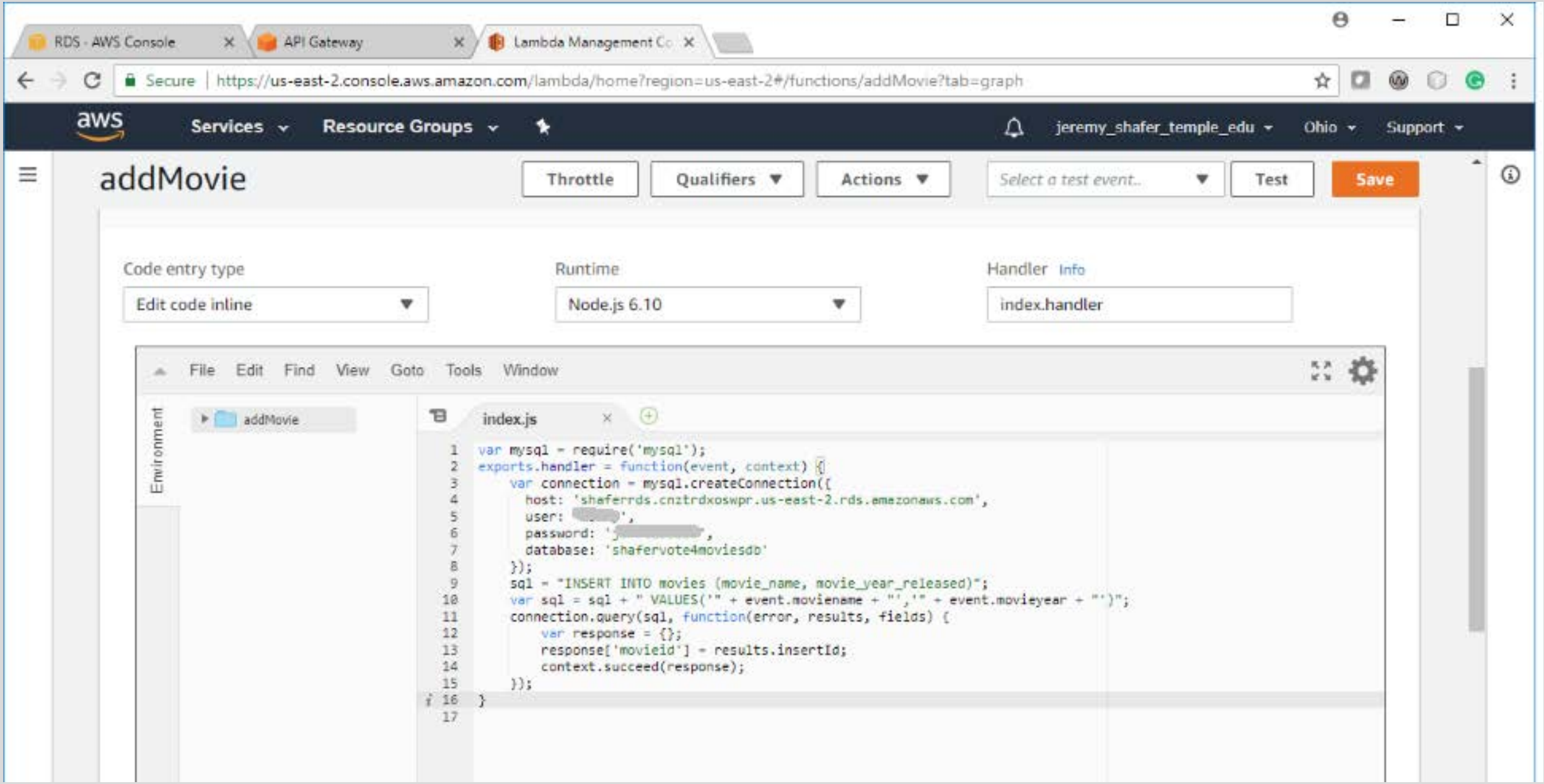
The language options were C#, Node.js, Python, and Go.

It was also necessary to specify the security role for each function.

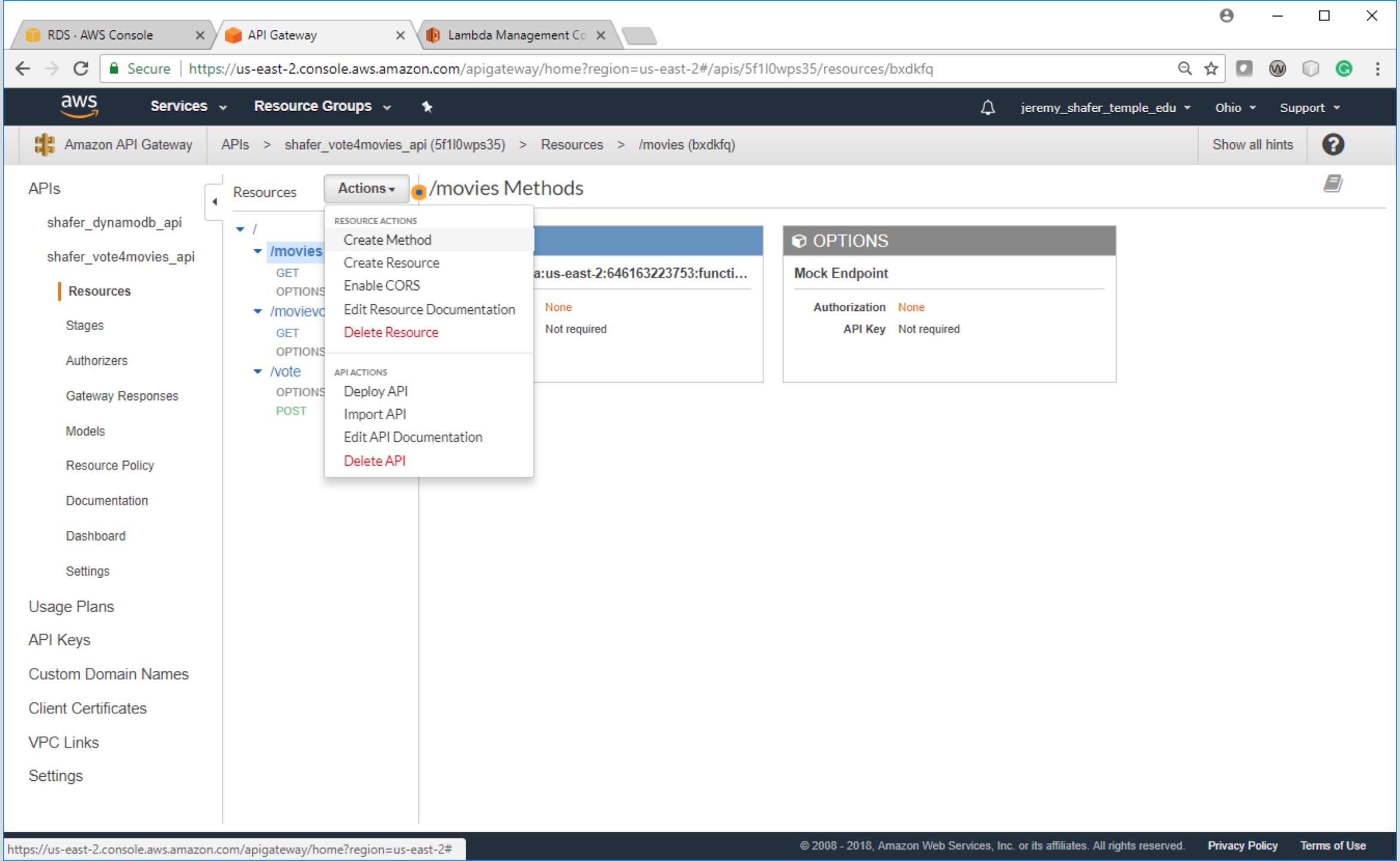
Getting started … note the ARN, as it will be needed later.

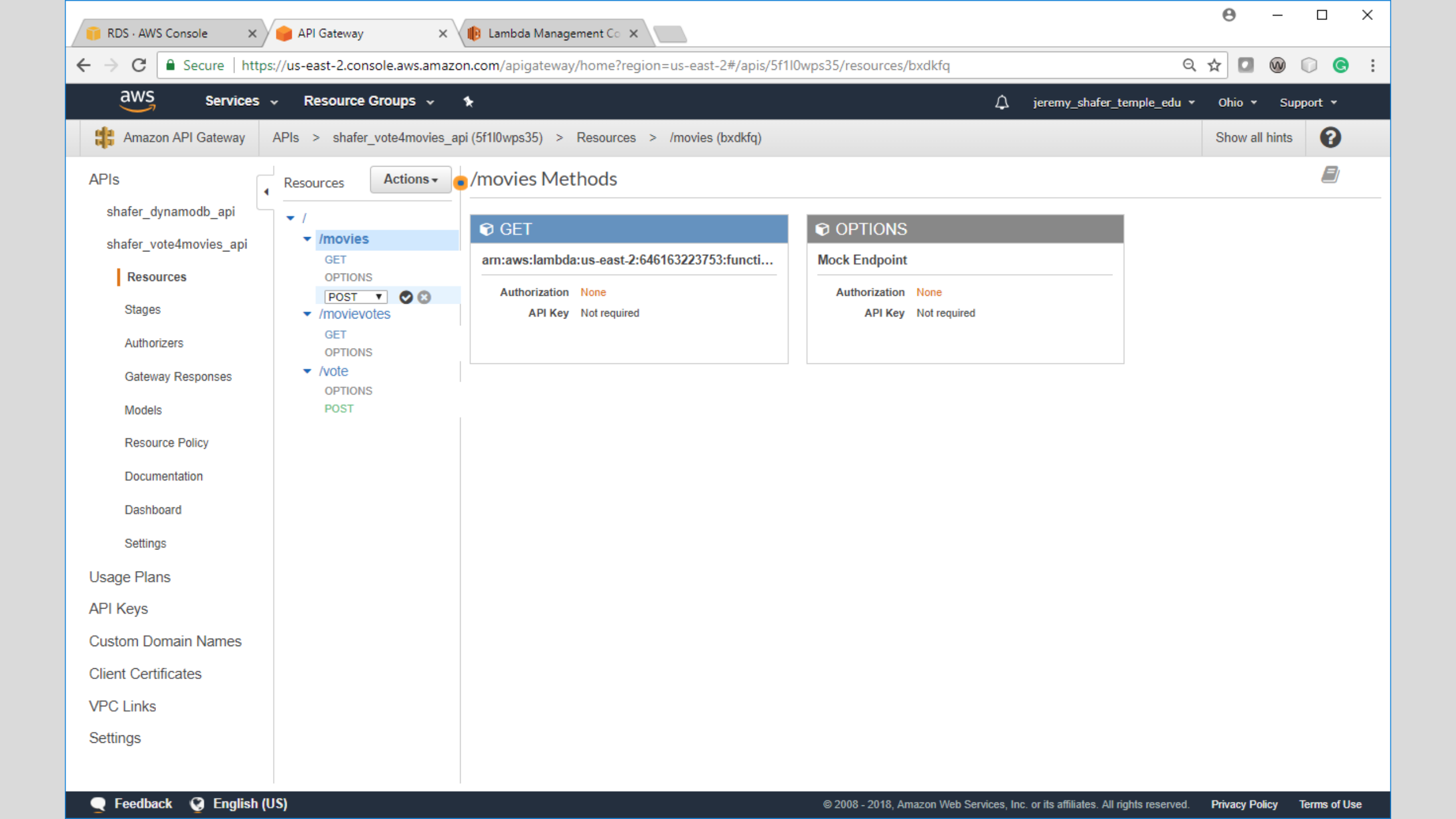


Code to be edited here. Each Lambda function needs to be tested.

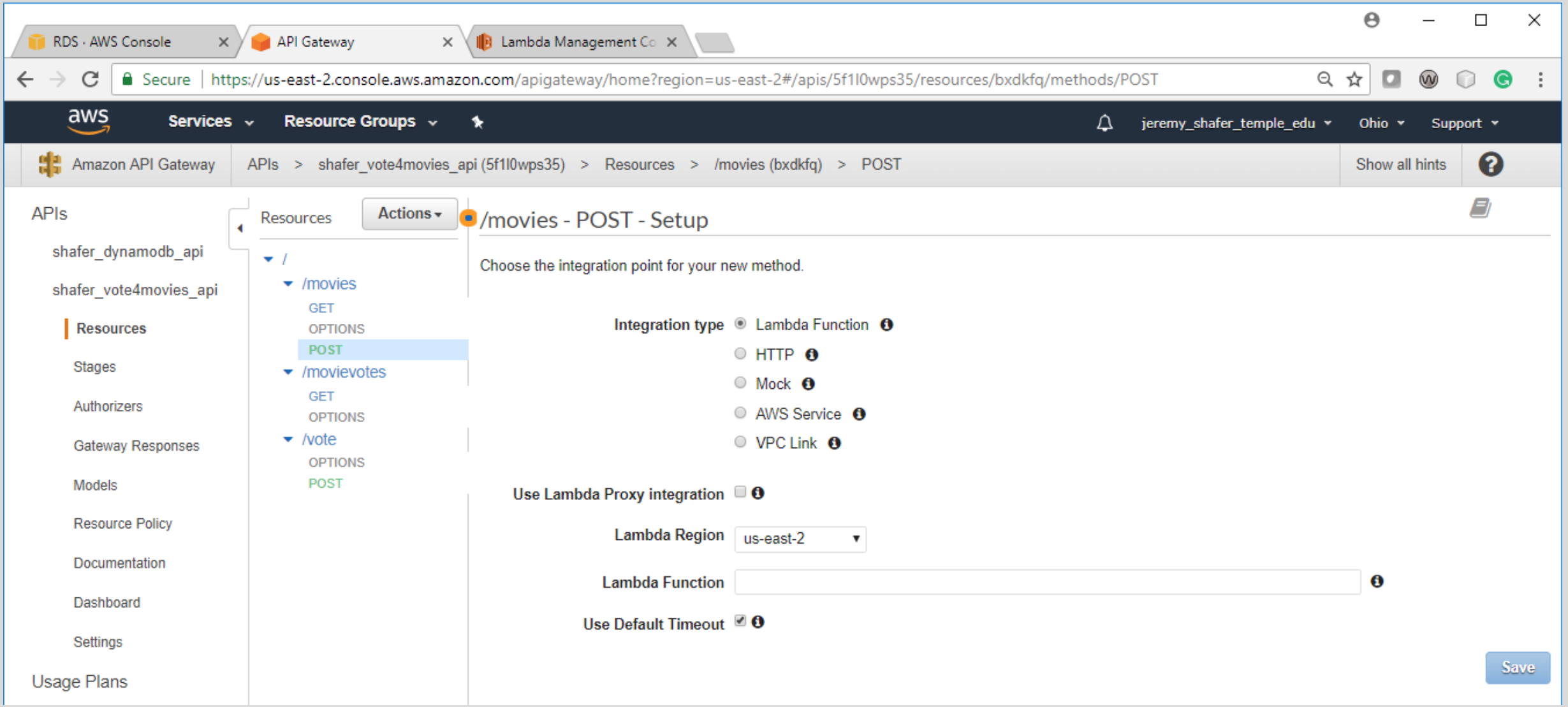


Define the Amazon API Gateway

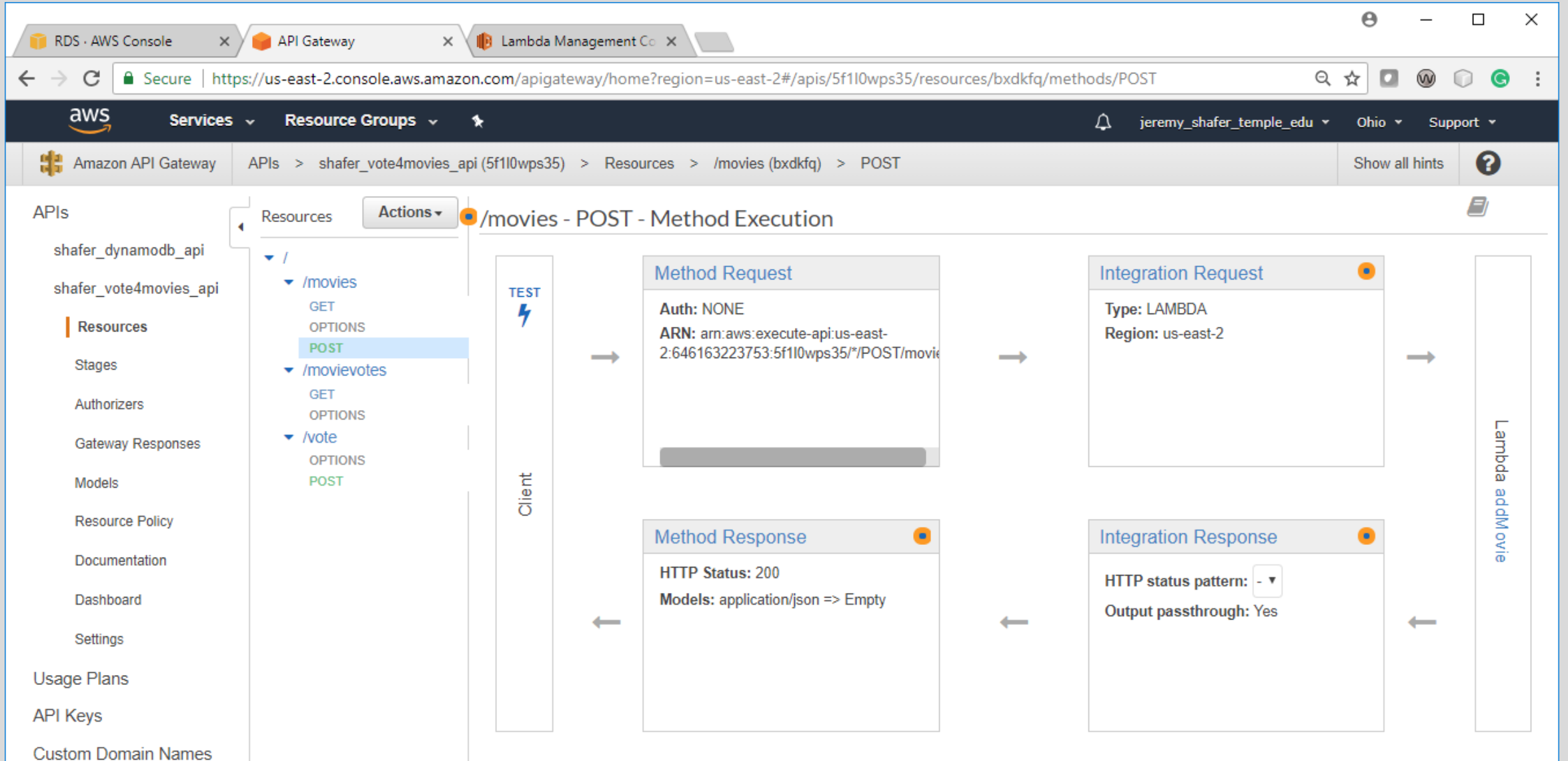

RDS - AWS Console
API Gateway
Lambda Management Co
Secure
https://us-east-2.console.aws.amazon.com/apigateway/home?region=us-east-2#/apis/5f1l0wps35/resources/bxdkfq
aws
Services
Resource Groups
jeremy_shafer_temple_edu
Ohio
Support
Amazon API Gateway
APIs > shafer_vote4movies_api (5f1l0wps35) > Resources > /movies (bxdkfq)
Show all hints
APIs
shafer_dynamodb_api
shafer_vote4movies_api
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Usage Plans
API Keys
Custom Domain Names
Client Certificates
VPC Links
Settings
Resources
Actions
/movies Methods
RESOURCE ACTIONS
Create Method
Create Resource
Enable CORS
Edit Resource Documentation
Delete Resource
API ACTIONS
Deploy API
Import API
Edit API Documentation
Delete API
/
/movies
GET
OPTIONS
/movievo
GET
OPTIONS
/vote
OPTIONS
POST
a:us-east-2:646163223753:functi...
None
Not required
OPTIONS
Mock Endpoint
Authorization None
API Key Not required
https://us-east-2.console.aws.amazon.com/apigateway/home?region=us-east-2#
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

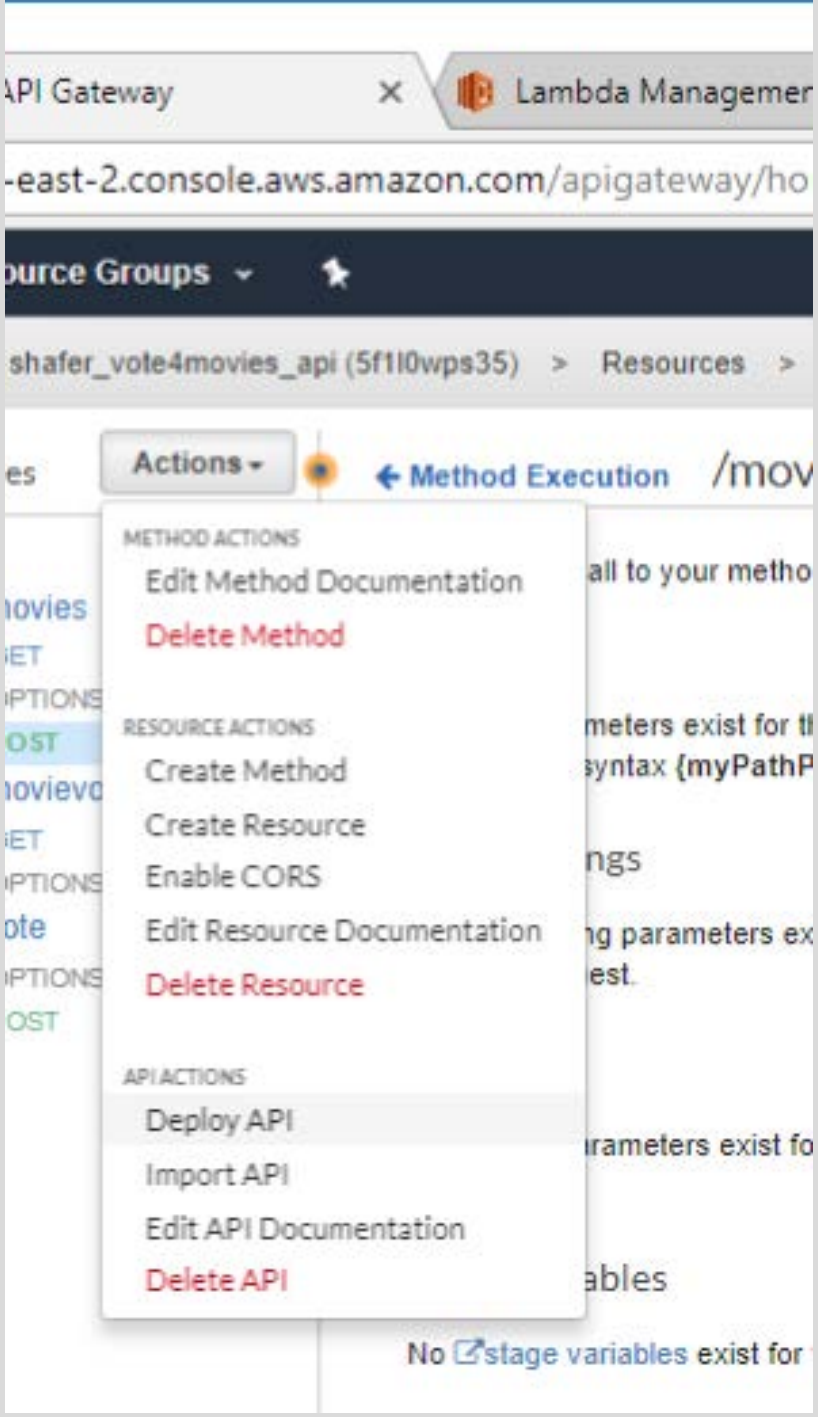

RDS - AWS Console
API Gateway
Lambda Management Co
Secure
https://us-east-2.console.aws.amazon.com/apigateway/home?region=us-east-2#/apis/5f1l0wps35/resources/bxdkfq
aws
Services
Resource Groups
jeremy_shafer_temple_edu
Ohio
Support
Amazon API Gateway
APIs > shafer_vote4movies_api (5f1l0wps35) > Resources > /movies (bxdkfq)
Show all hints
APIs
shafer_dynamodb_api
shafer_vote4movies_api
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Dashboard
Settings
Usage Plans
API Keys
Custom Domain Names
Client Certificates
VPC Links
Settings
Resources
Actions
/movies Methods
/
/movies
GET
OPTIONS
POST
/movievotes
GET
OPTIONS
/vote
OPTIONS
POST
GET
arn:aws:lambda:us-east-2:646163223753:functi...
Authorization None
API Key Not required
OPTIONS
Mock Endpoint
Authorization None
API Key Not required
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Now we integrate our new Lambda function with a web API.



The Amazon API Gateway gives a visual representation of the web call and response.





Now deploy the API

The results

Four Endpoints

Simple GET request: Gets all the movies - returns a JSON array
Example: {"movie_id":22,"movie_name":"The Avengers","movie_year_released":"2012"}
https://5f1l0wps35.execute-api.us-east-2.amazonaws.com/shafer_vote4movies/movies

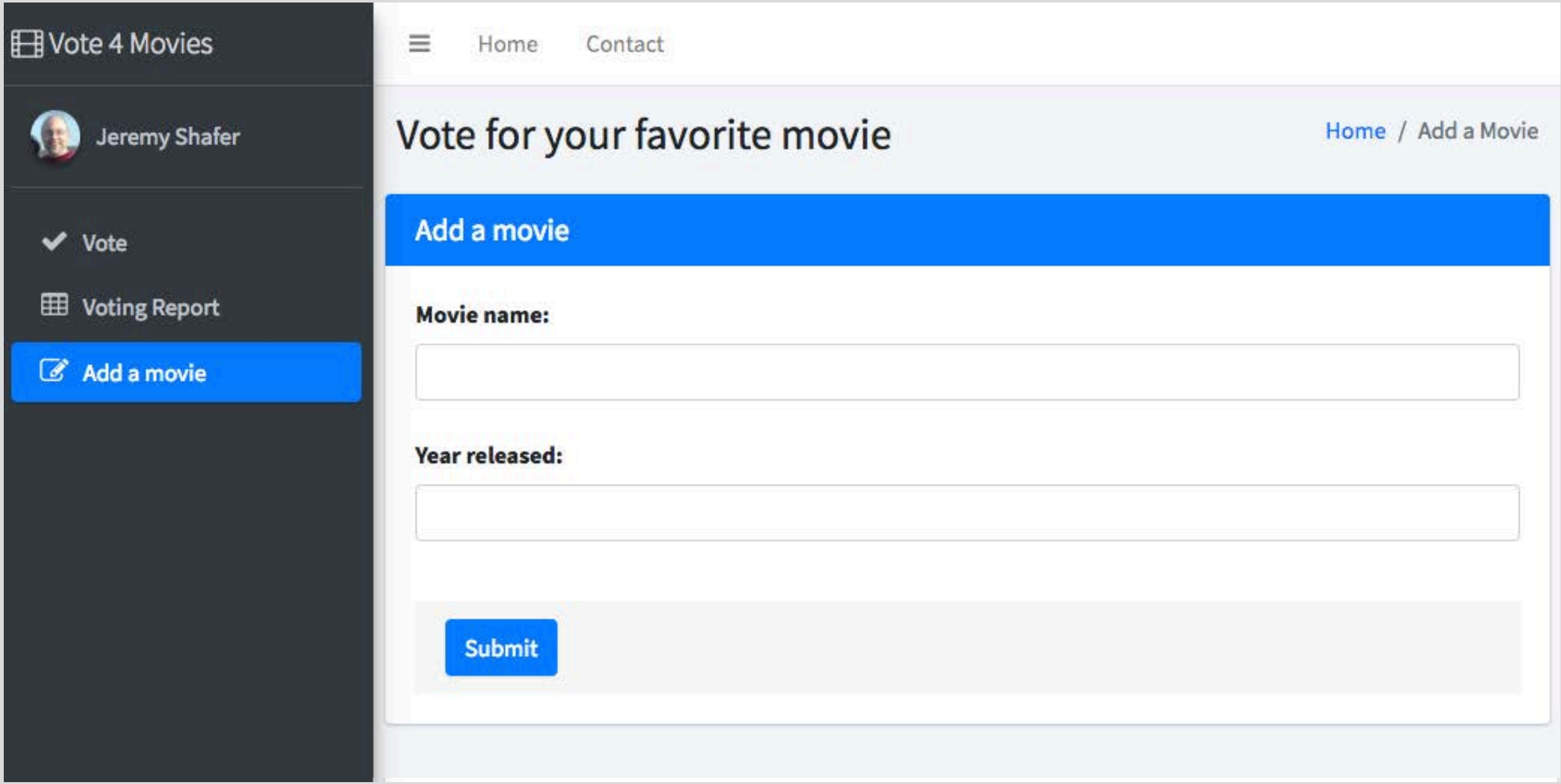
Simple GET request: Gets movie names and the number of votes - returns a JSON array
Example: {"movie_id":22,"movie_name":"The Avengers","movie_year_released":"2012"}

https://5f1l0wps35.execute-api.us-east-2.amazonaws.com/shafer_vote4movies/movievotes

Vote for a movie.
This is a POST. POST the following JSON {"movieid":"22"} (Here the 22 corresponds to "The Avengers")
Returns the new vote_id and the original movie_id
https://5f1l0wps35.execute-api.us-east-2.amazonaws.com/shafer_vote4movies/vote

Add a movie.
This is a POST. POST the following JSON {"moviename":"E.T.", "movieyear":"1982"}
The api returns the key of the movie just created.
https://5f1l0wps35.execute-api.us-east-2.amazonaws.com/shafer_vote4movies/movies

Now it’s time to build the web application.


Vote 4 Movies
Jeremy Shafer
Vote
Voting Report
Add a movie
Home
Contact
Vote for your favorite movie
Home / Add a Movie
Add a movie
Movie name:
Year released:
Submit