Serverless Application Development on AWS

Bob Reselman

Please, take the pre-session survey:

https://www.surveymonkey.com/r/Q699JMF

Brief Bio


InformationWeek
NETWORKComputing


DevOps.com


Gateway™


THE RECORDING ACADEMY


edmunds

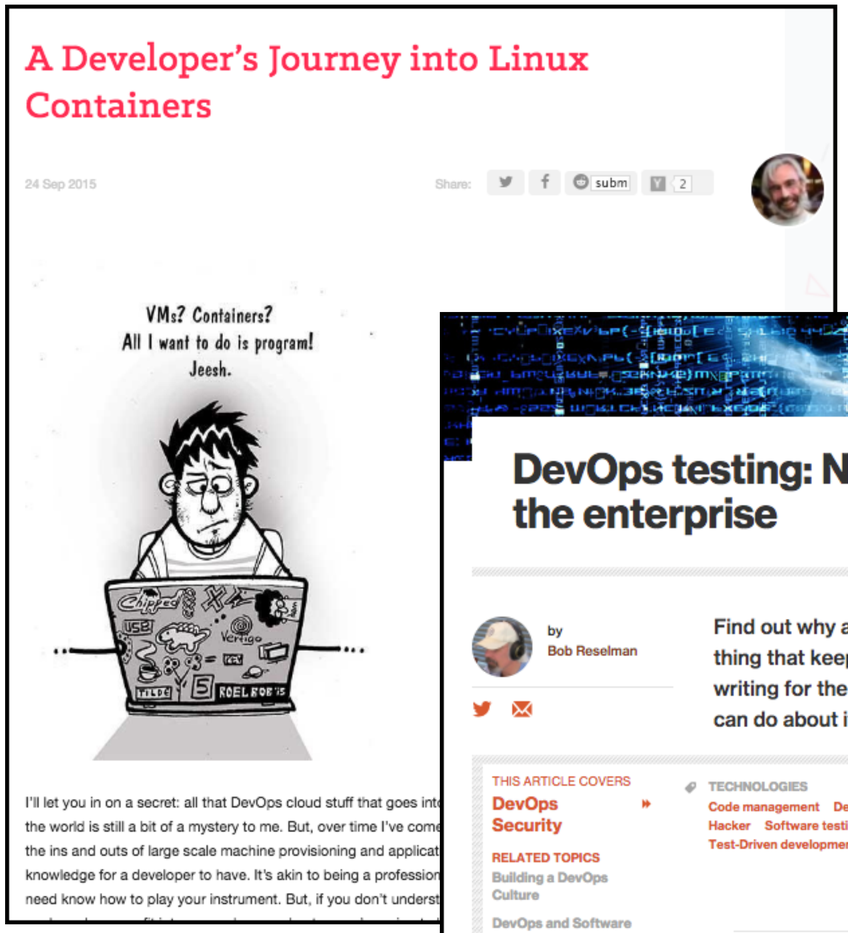

SIMPSON
COLLEGE


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING


logentries™

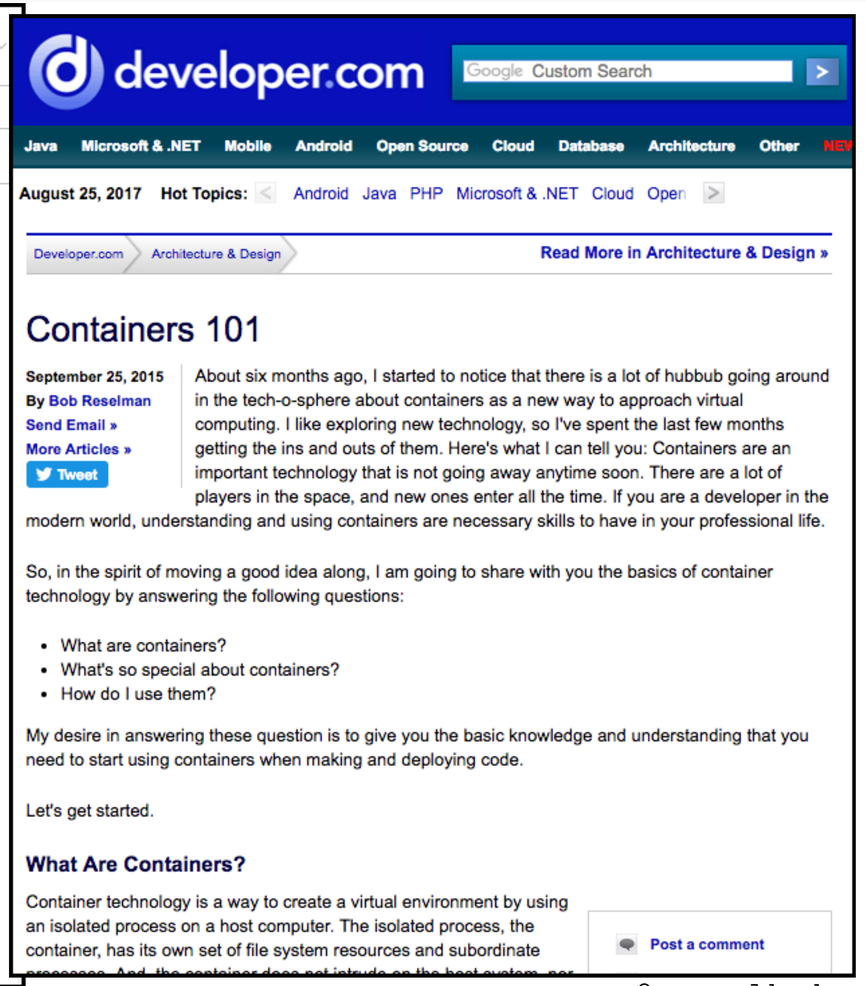

Casting Networks
INTERNATIONAL

Brief Bio


A Developer's Journey into Linux Containers
24 Sep 2015
Share:
VMs? Containers?
All I want to do is program!
Jeesh.


DevOps testing: Never trust the world outside the enterprise
by
Bob Reselman
Find out why a white-hat hacker claims "the biggest thing that keeps me up at night is the code DevOps is writing for the infrastructure," and what security pros can do about it.
THIS ARTICLE COVERS
DevOps Security
RELATED TOPICS
Building a DevOps Culture
DevOps and Software Development
TECHNOLOGIES
Code management DevOps
Hacker Software testing & QA
Test-Driven development

ProgrammableWeb
API NEWS
API DIRECTORY
LEARN ABOUT APIS
WHAT IS AN API ?
API RESEARCH
WEATHER
MAPPING
Why Messaging Queues Suck
API UNIVERSITY
Analysis, Integration


developer.com
Google Custom Search
Java
Microsoft & .NET
Mobile
Android
Open Source
Cloud
Database
Architecture
Other
August 25, 2017
Hot Topics:
Android Java PHP Microsoft & .NET Cloud
Developer.com
Architecture & Design
Read More in Architecture & Design »
Containers 101
September 25, 2015
By Bob Reselman
Send Email »
More Articles »
Tweet
About six months ago, I started to notice that there is a lot of hubbub going around in the tech-o-sphere about containers as a new way to approach virtual computing. I like exploring new technology, so I've spent the last few months getting the ins and outs of them. Here's what I can tell you: Containers are an important technology that is not going away anytime soon. There are a lot of players in the space, and new ones enter all the time. If you are a developer in the modern world, understanding and using containers are necessary skills to have in your professional life.
So, in the spirit of moving a good idea along, I am going to share with you the basics of container technology by answering the following questions:
What are containers?
What's so special about containers?
How do I use them?
My desire in answering these question is to give you the basic knowledge and understanding that you need to start using containers when making and deploying code.
Let's get started.
What Are Containers?
Container technology is a way to create a virtual environment by using an isolated process on a host computer. The isolated process, the container, has its own set of file system resources and subordinate
Post a comment

Agenda for Day 1

- **Session 1**
  - Setting Up An AWS Account
  - Creating the Class User & Assigning Permissions
  - Creating the Static Web Site on S3
- **Session 2**
  - Create Class Role
  - Create Lambda
  - View in CloudWatch
- **Session 3**
  - Create API Gateway Endpoints
  - Bind to Lambda
  - Testing the API
  - Creating the DynamoDB Table

Agenda for Day 2

- **Session 4**
  - Creating the POST Rating API Endpoint
  - Creating the postRating Lambda Function
  - Creating the Rating Index in DynamoDB
- **Session 5**
  - Creating the getMovies Lambda Function
  - Refactor the GET Movies API Endpoint
  - Creating the getRating Lambda Function
  - Creating the GET Ratings API Endpoint
  - Publishing the API to a Test Deployment
- **Session 6**
  - Refactoring the Movie Rater UI
  - Creating and Binding the SNS Topic
  - Subscribing to the MovieRater Topic

Preliminaries

GitHub Repo for Project

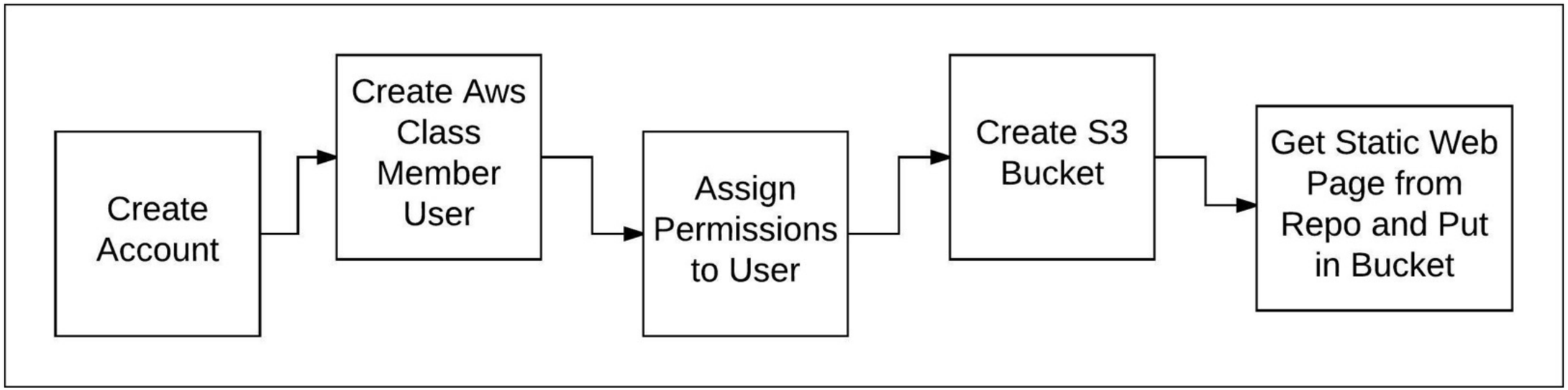
https://github.com/reselbob/CDAwsClass

API for IMDB Movie Data

https://www.omdbapi.com/

Sign up and get an API key

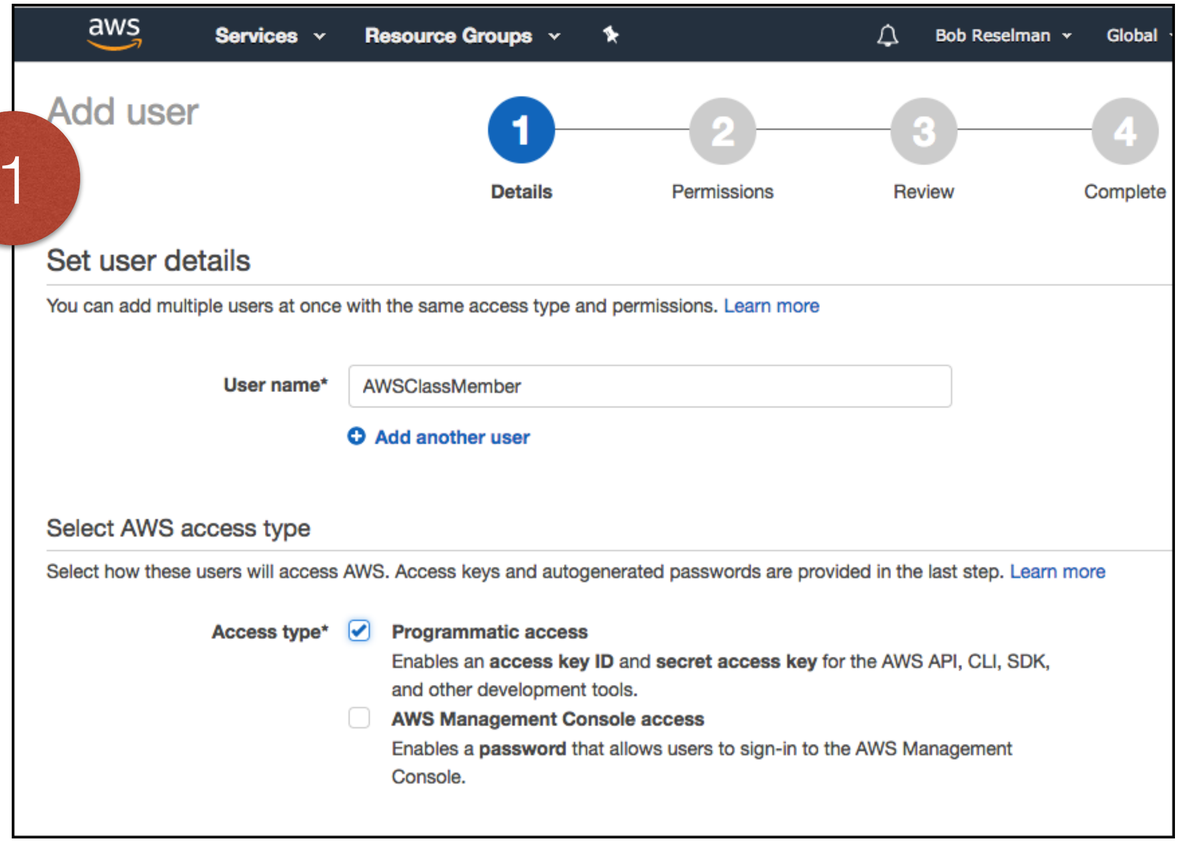
Session 1


Create Account
Create Aws Class Member User
Assign Permissions to User
Create S3 Bucket
Get Static Web Page from Repo and Put in Bucket

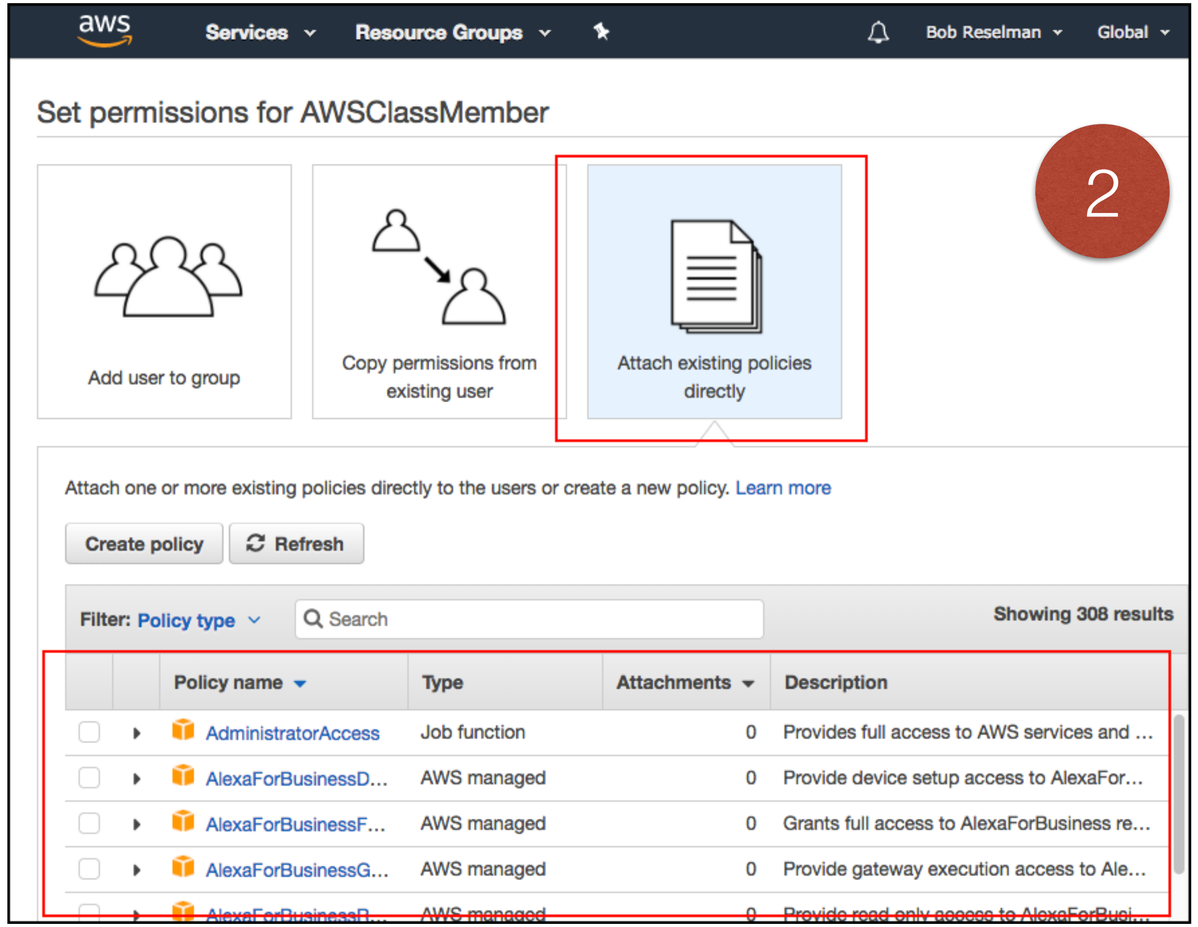
Setting Up An AWS Account

Creating the Class User & Assigning Permissions

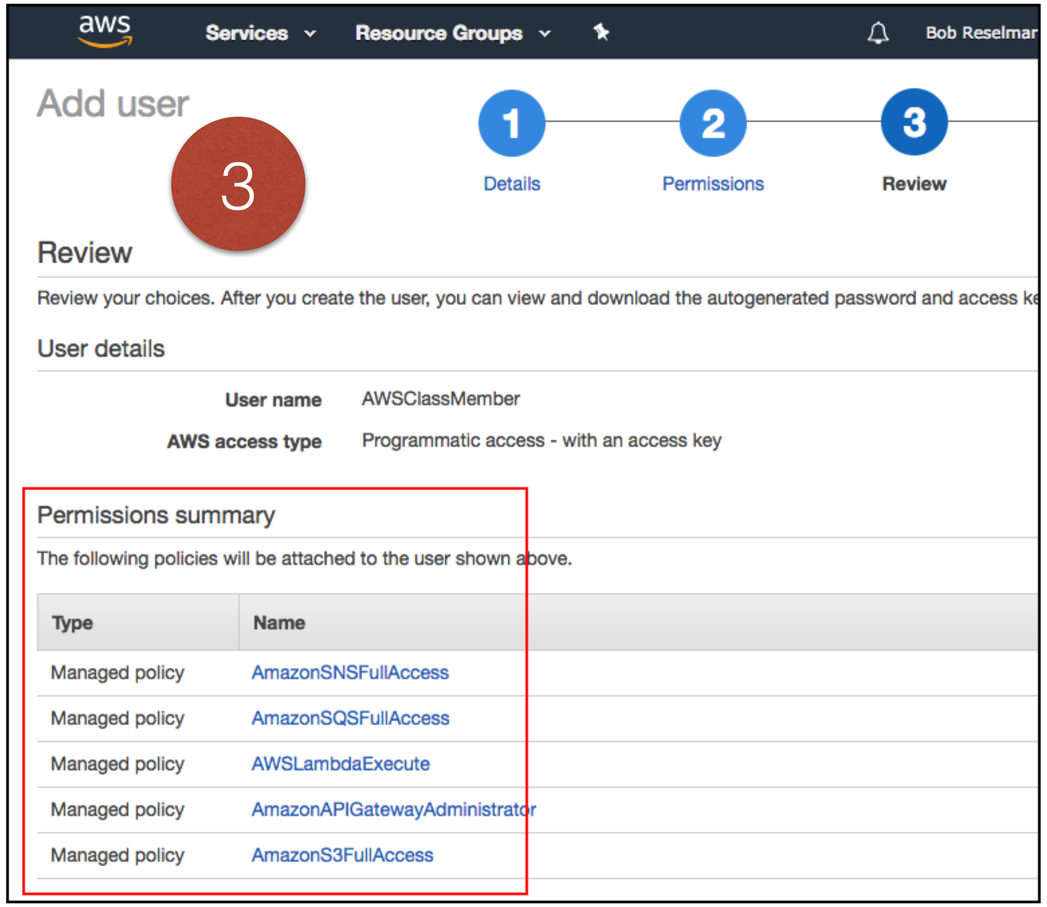
1

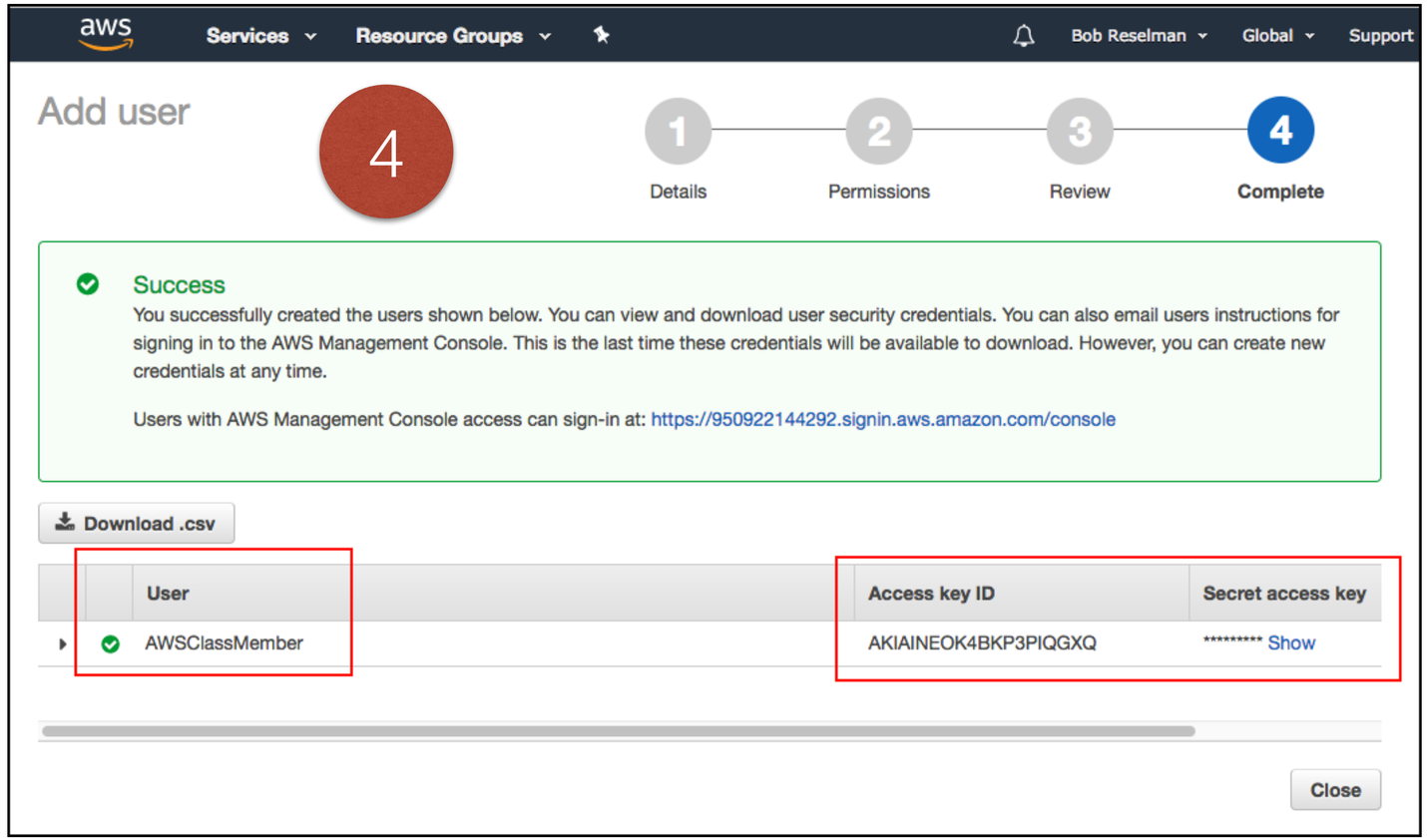


2

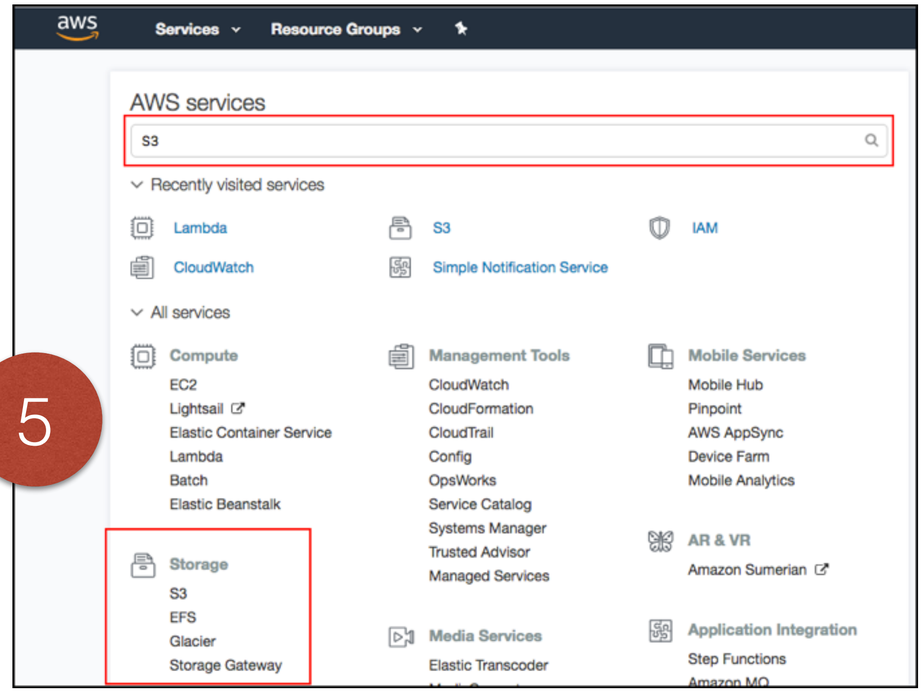


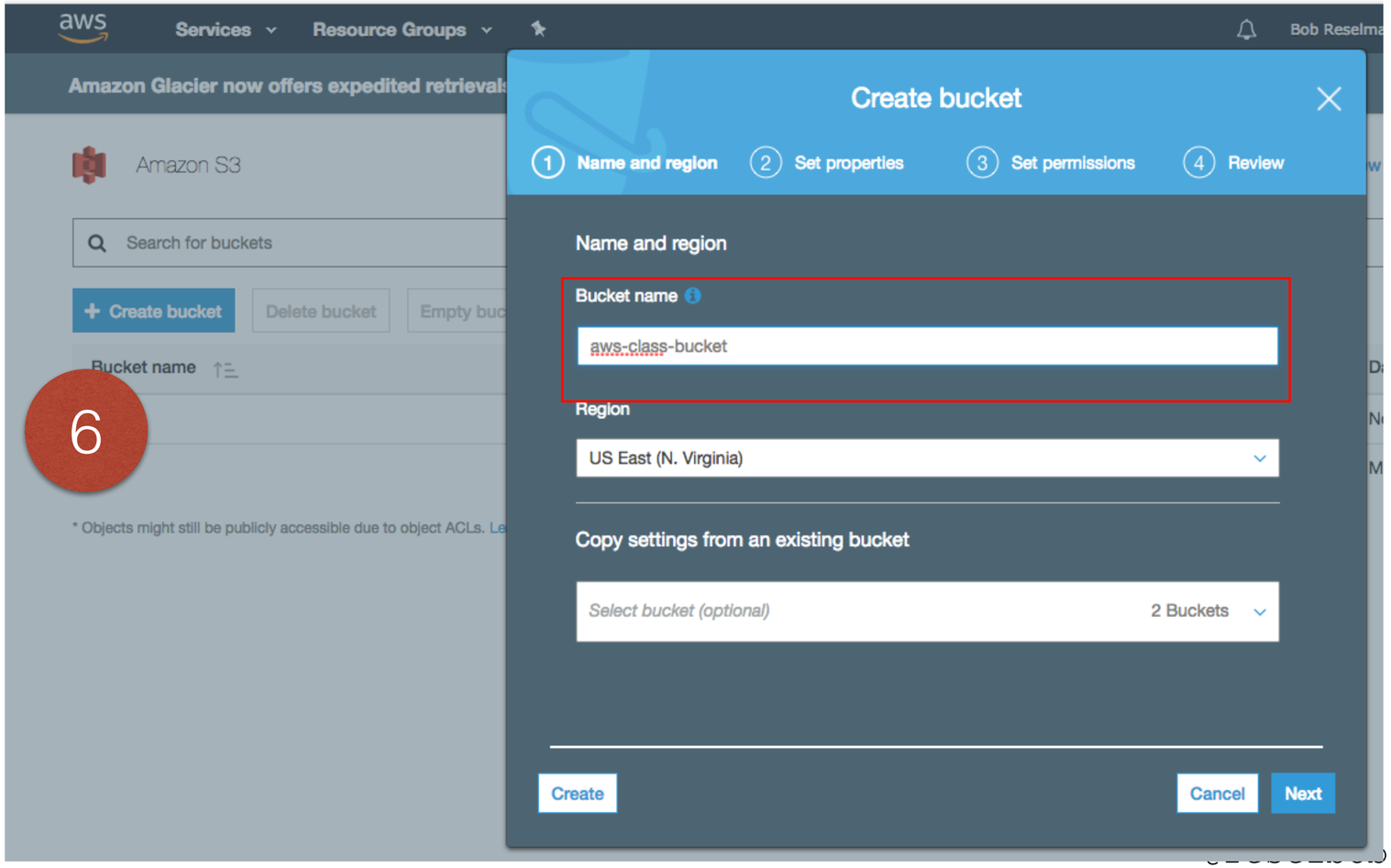
Creating the Class User & Assigning Permissions



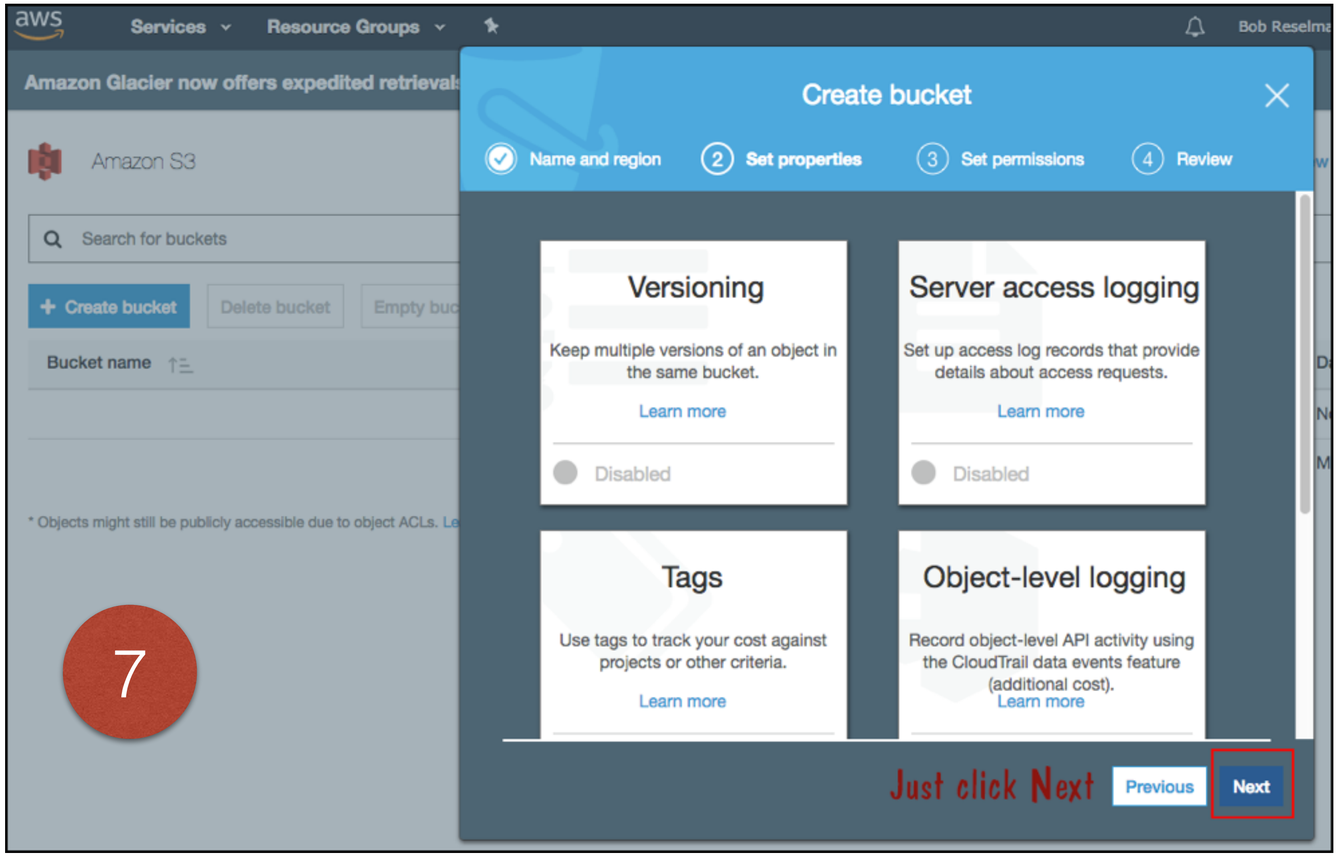


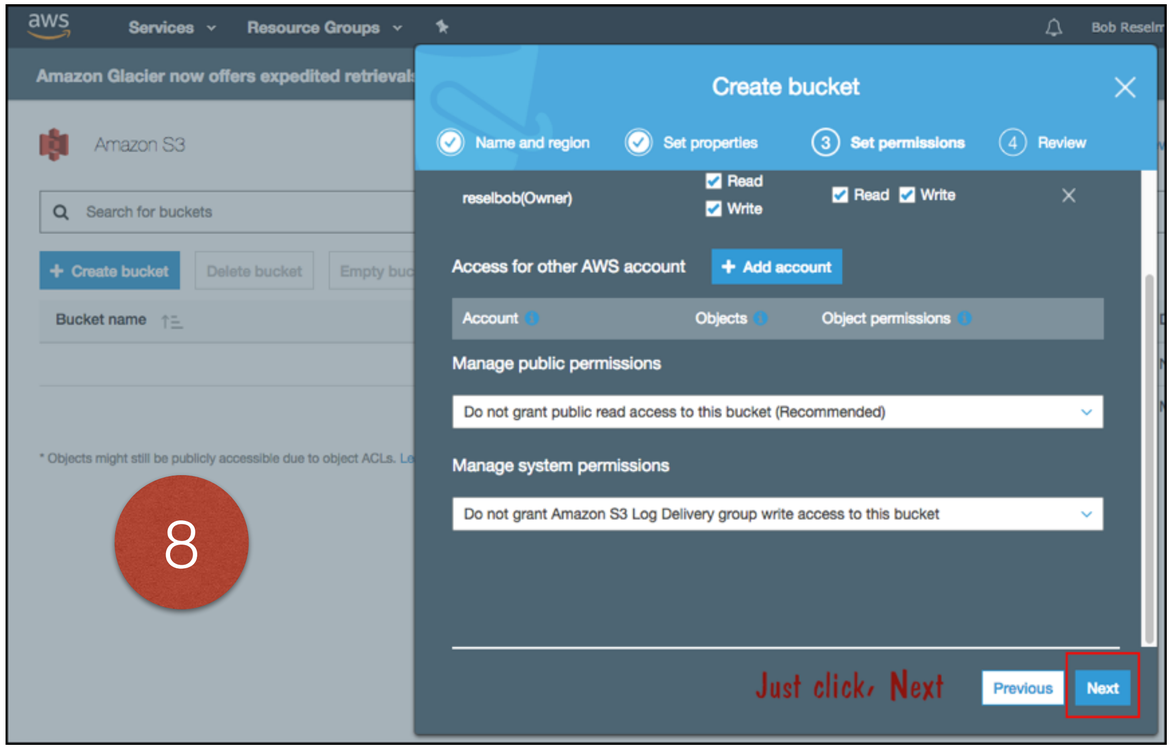
Create the S3 Bucket



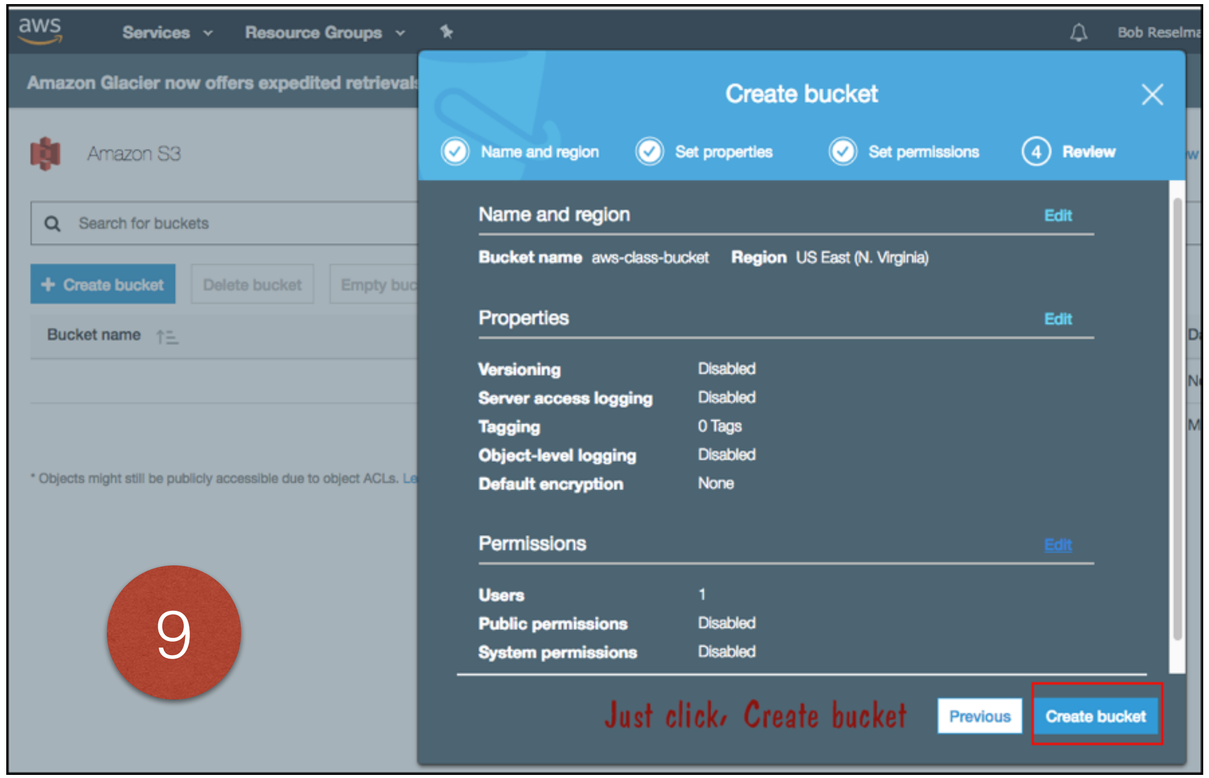


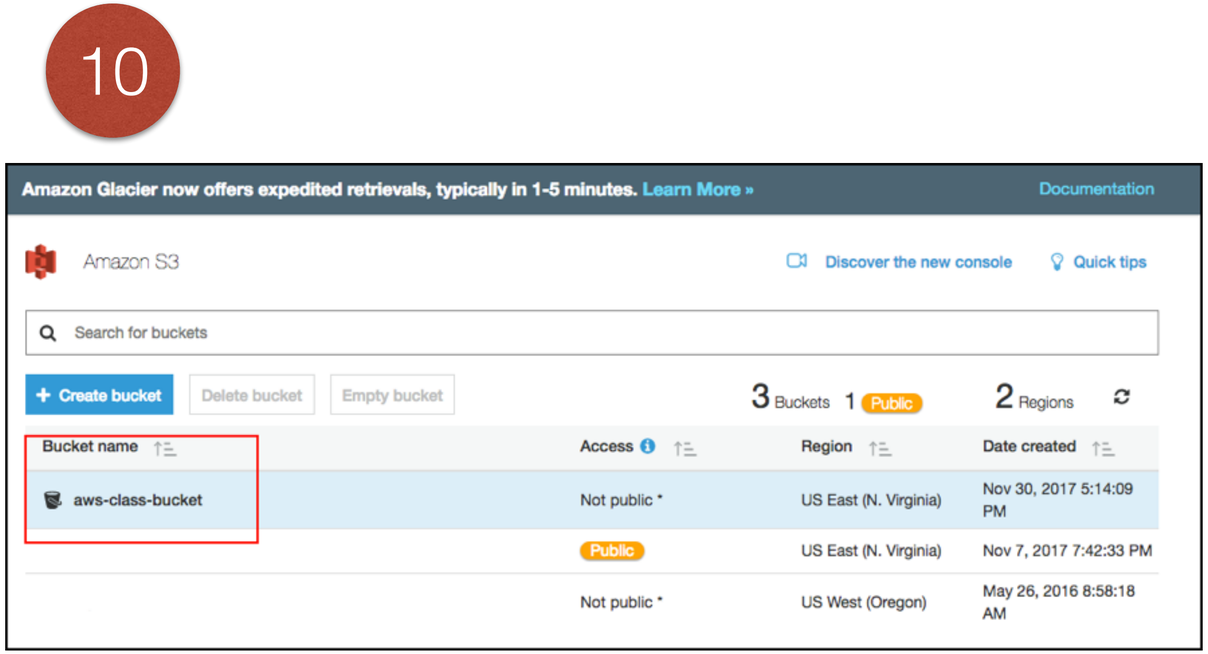
Create the S3 Bucket



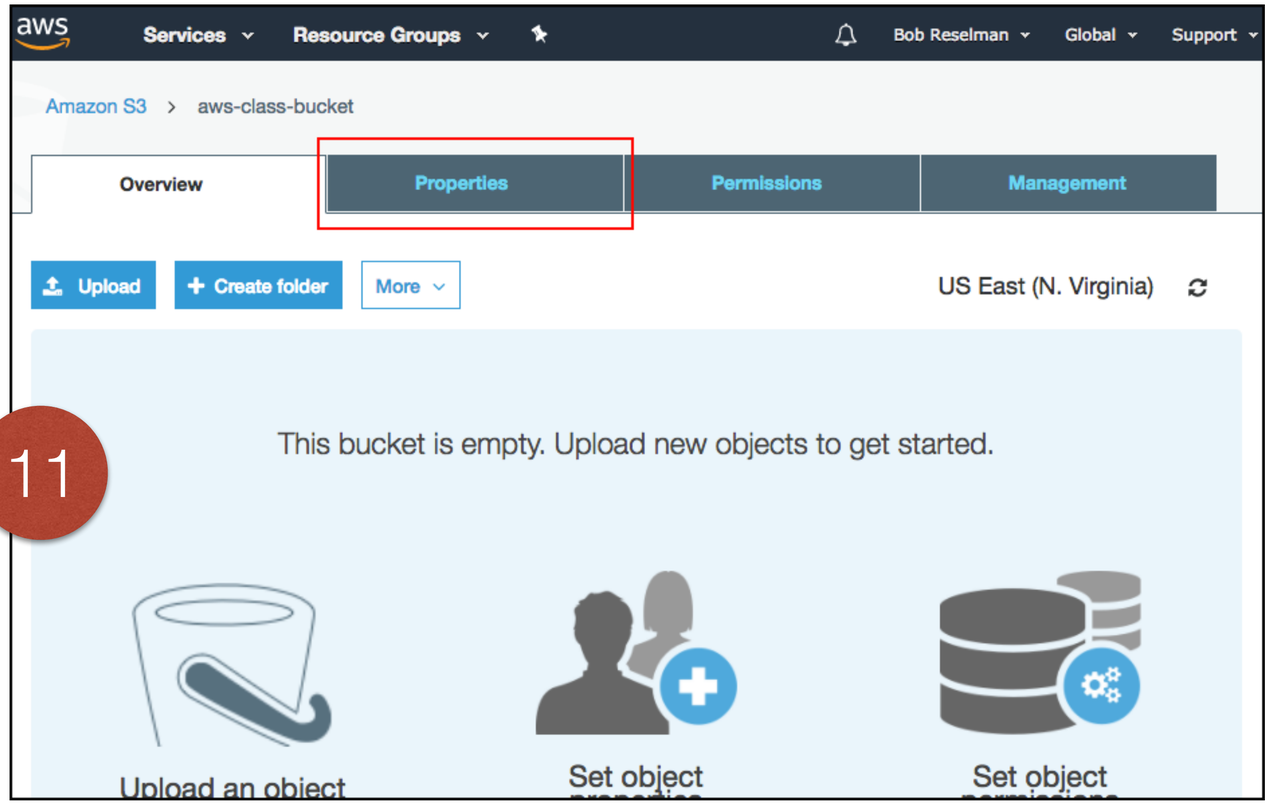


Create the S3 Bucket

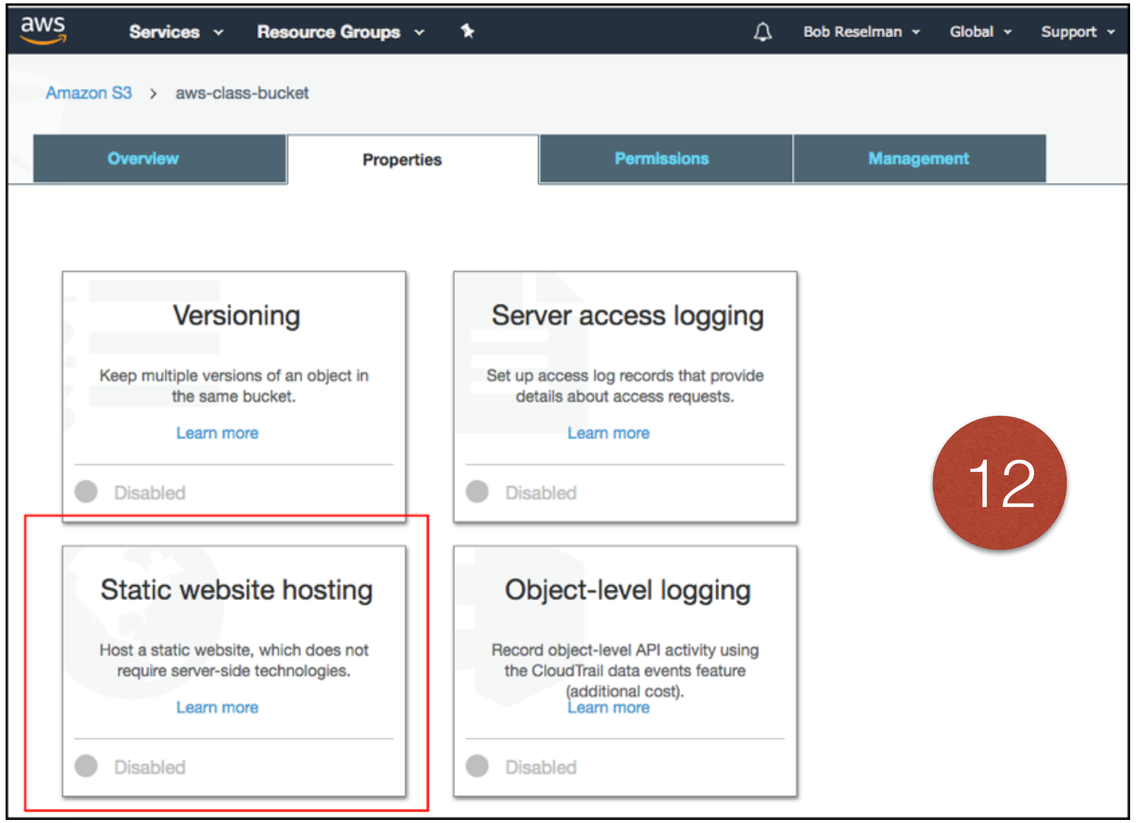




Create the S3 Bucket

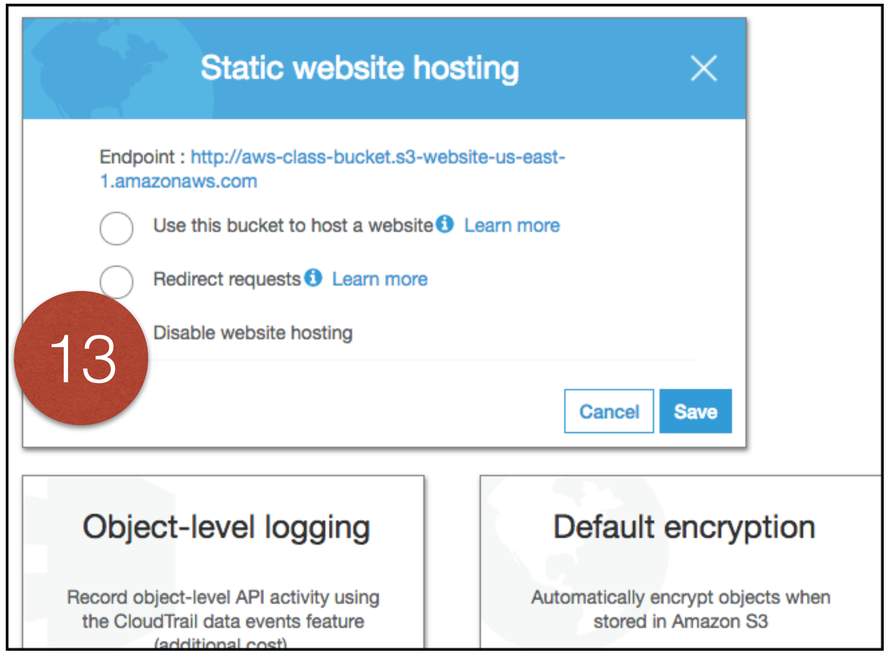


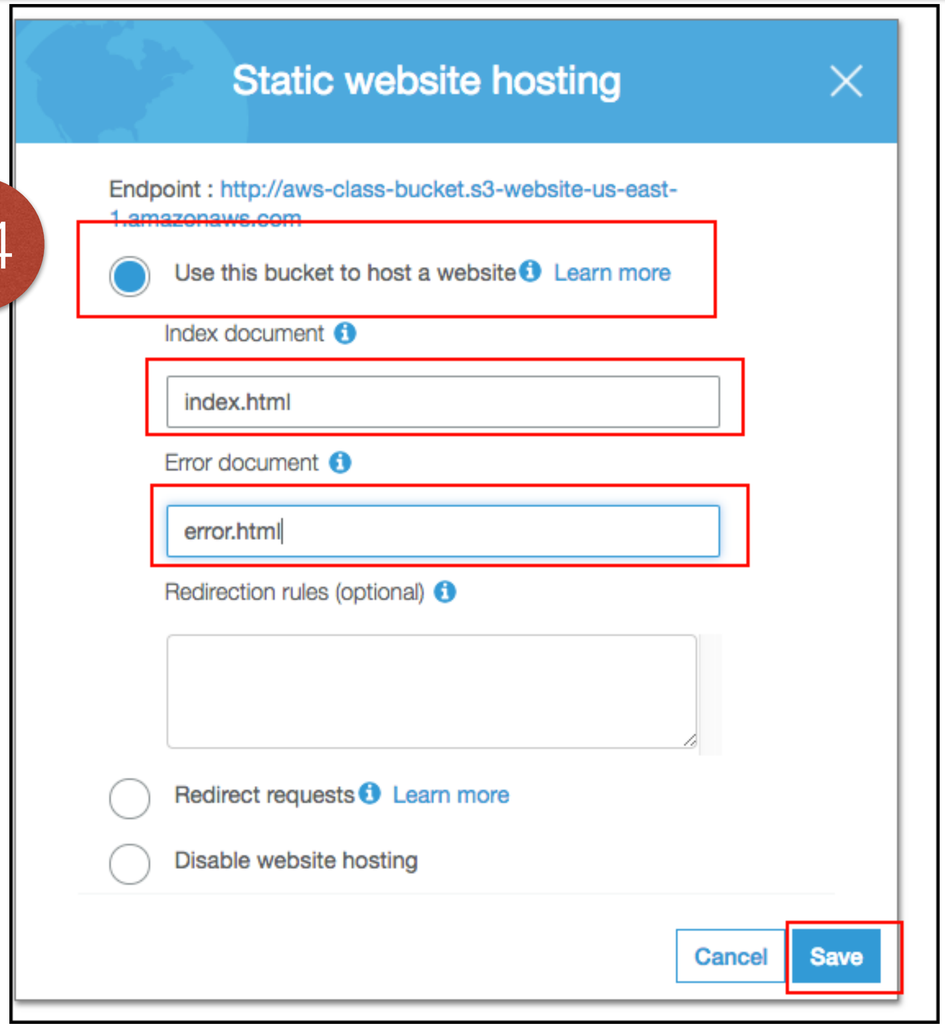
11



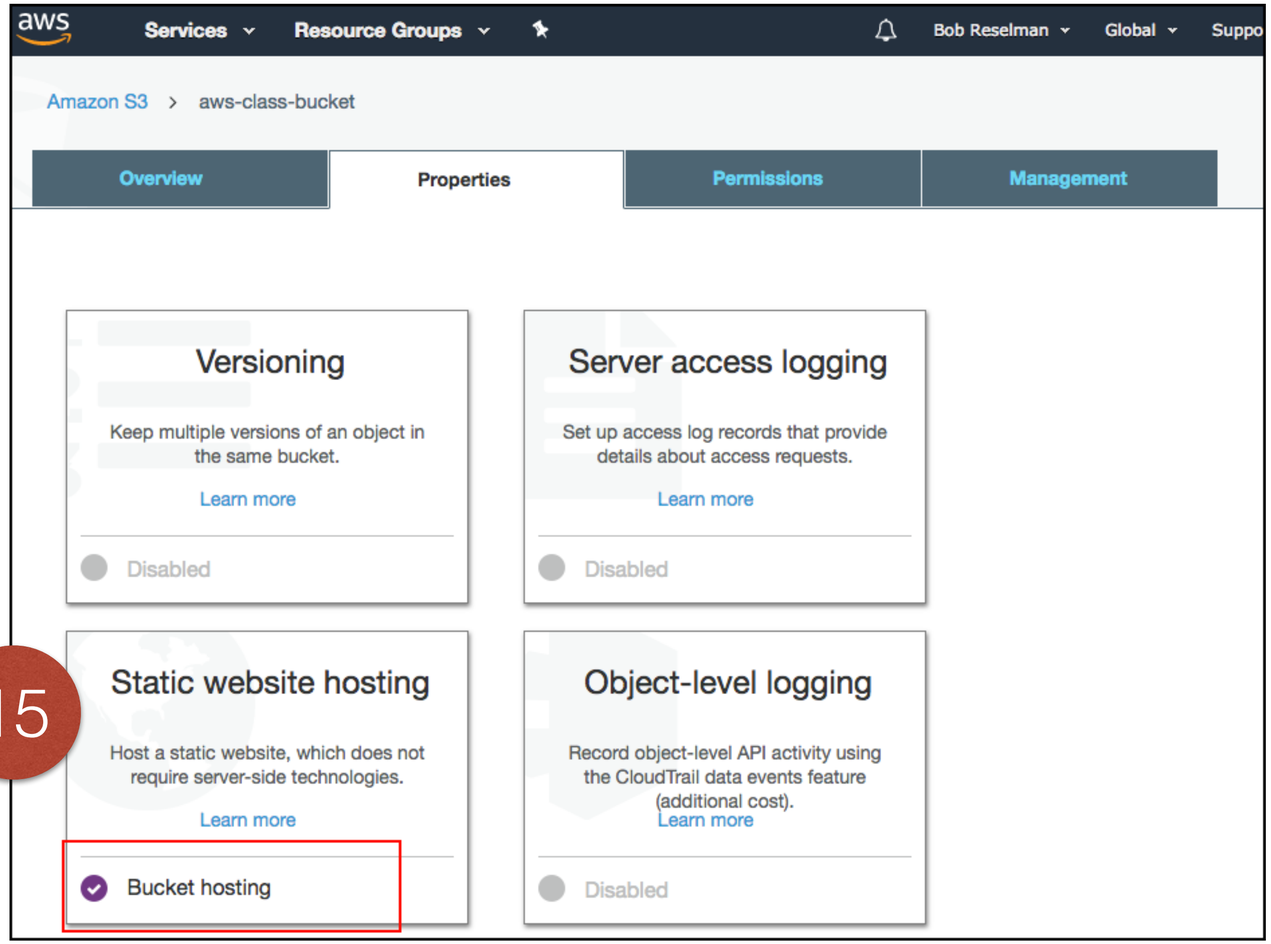
12

Create the S3 Bucket



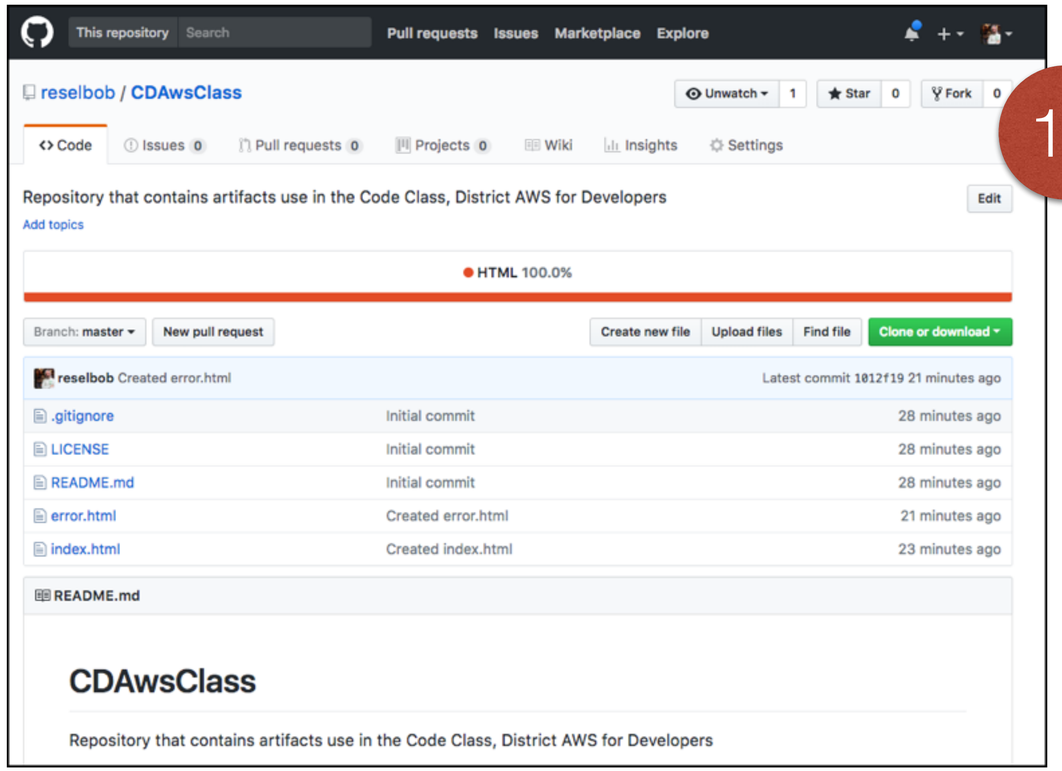


Create the S3 Bucket



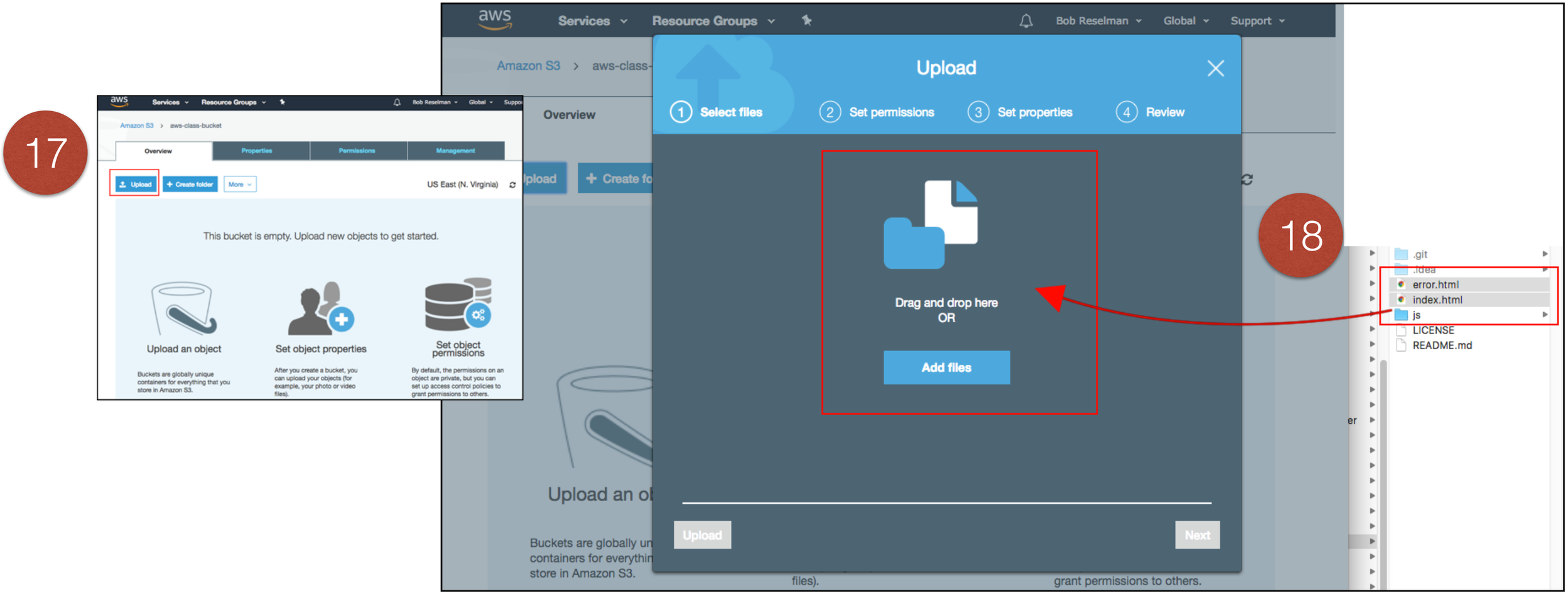
15

https://github.com/reselbob/CDAwsClass

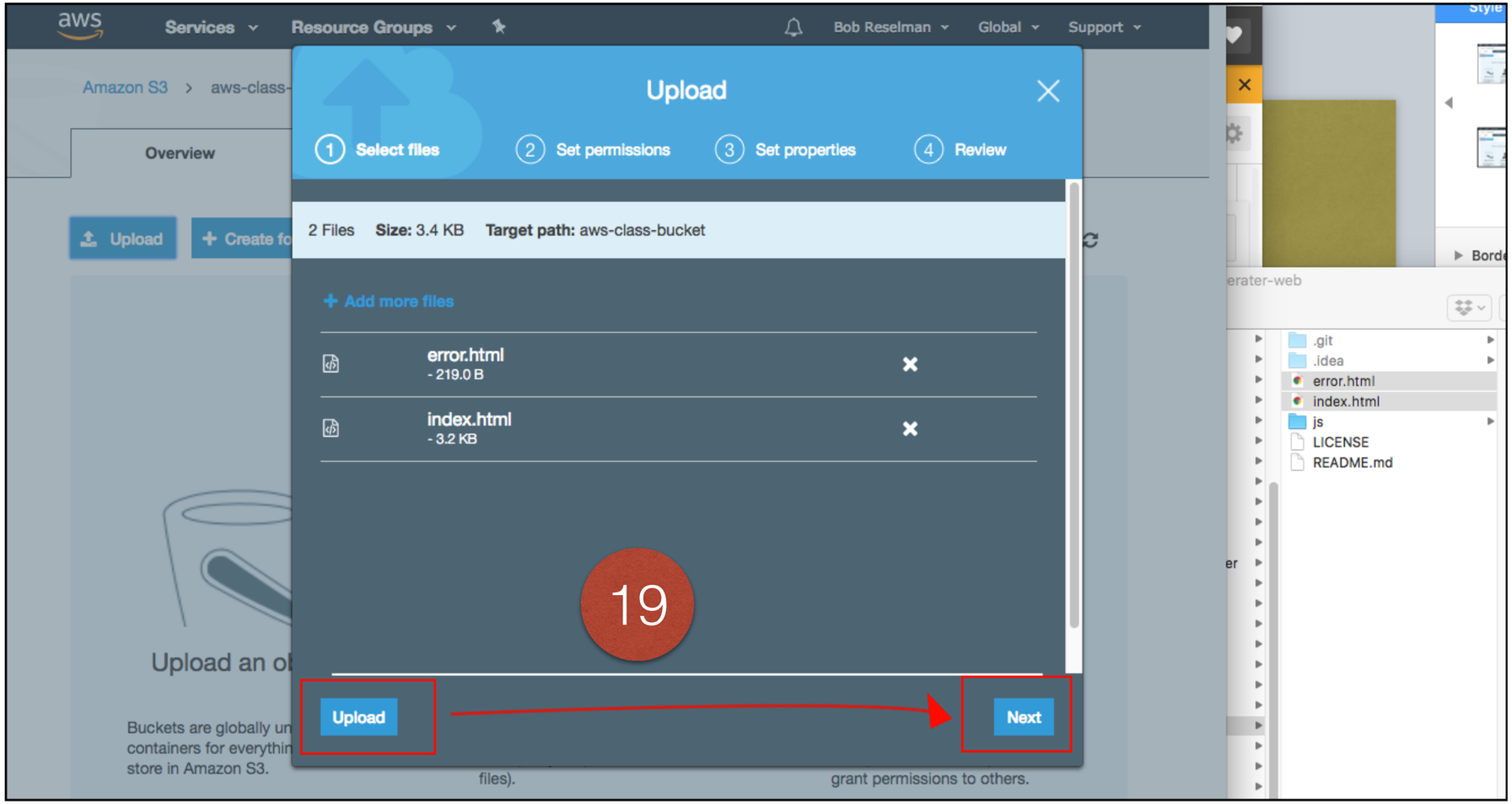


16

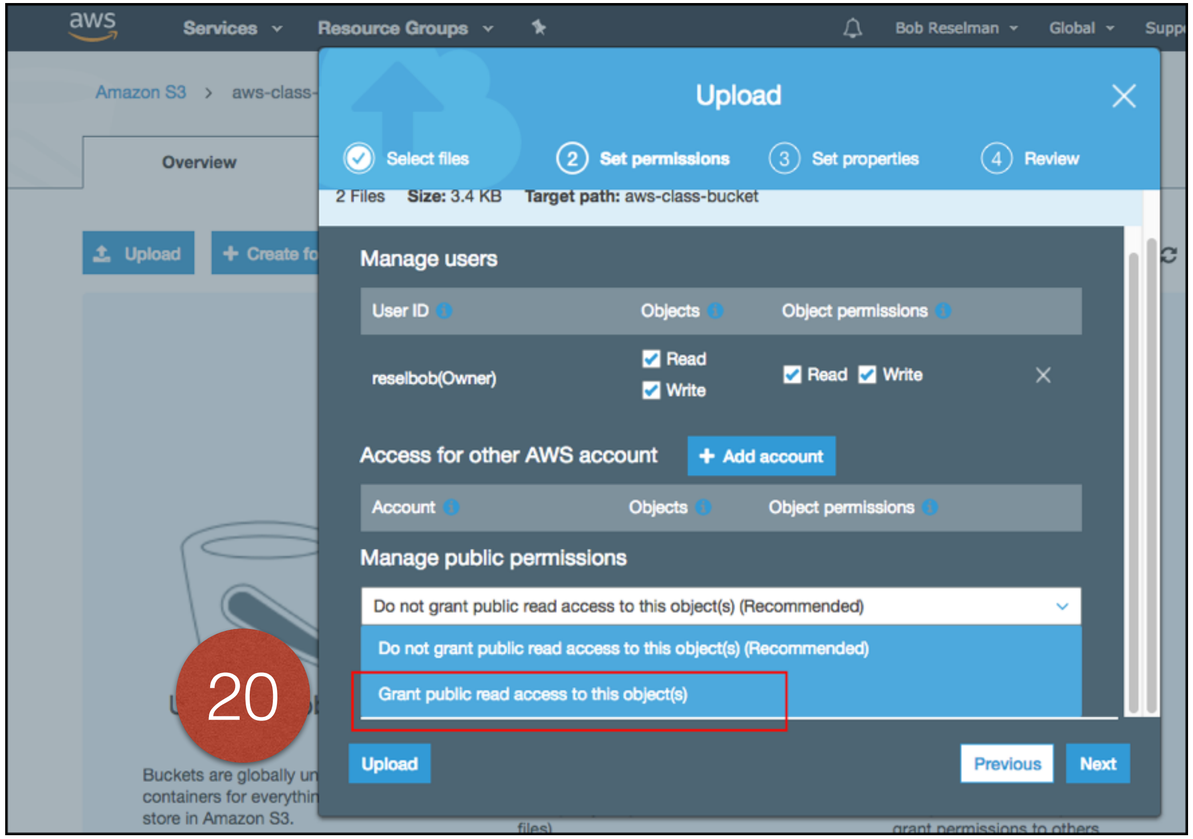
Create the S3 Bucket

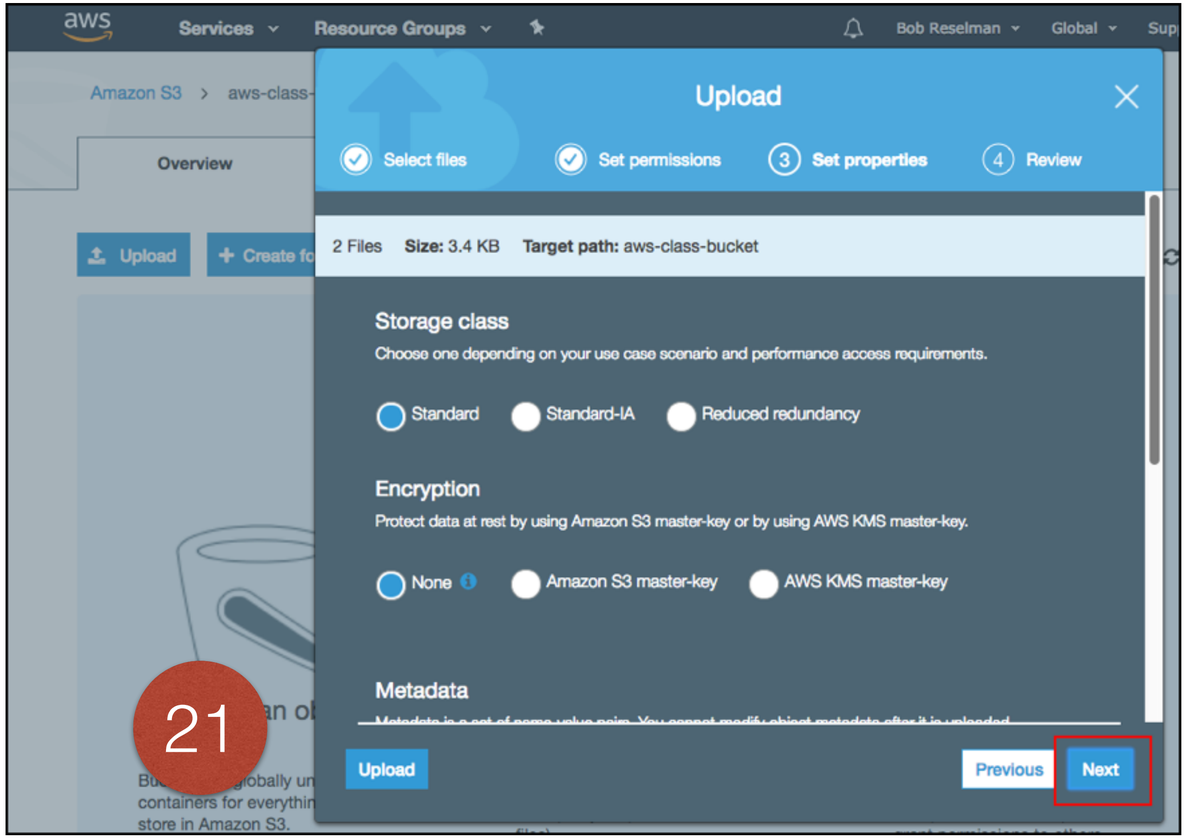


Create the S3 Bucket

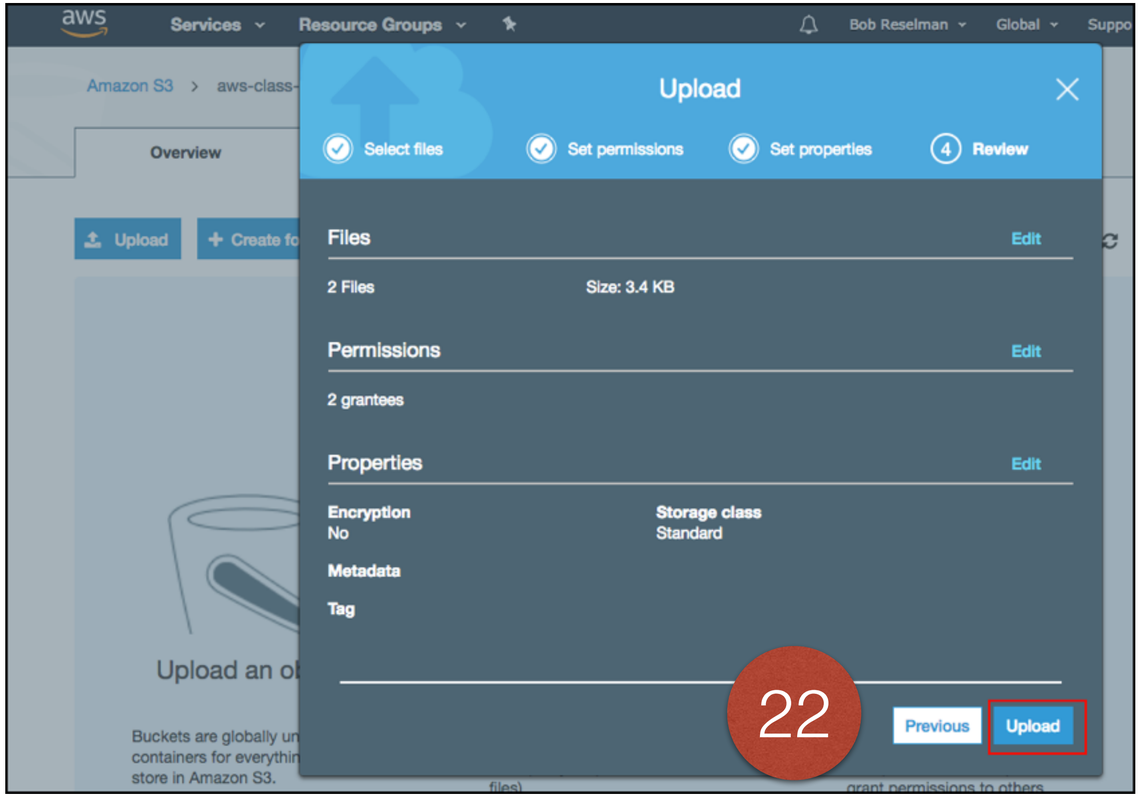


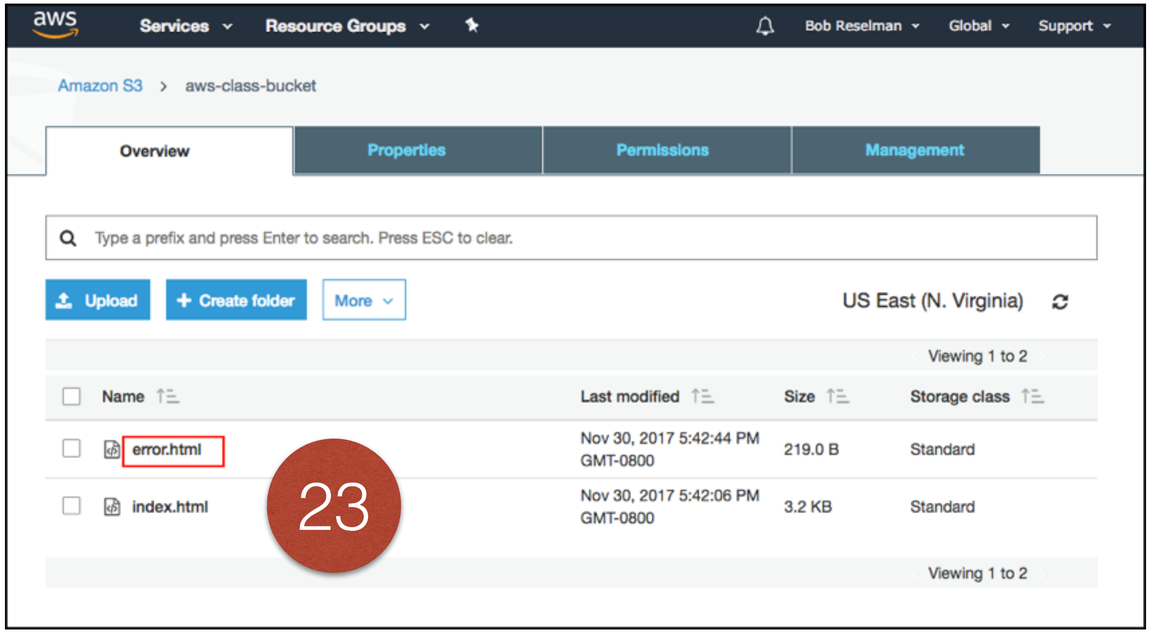
Create the S3 Bucket





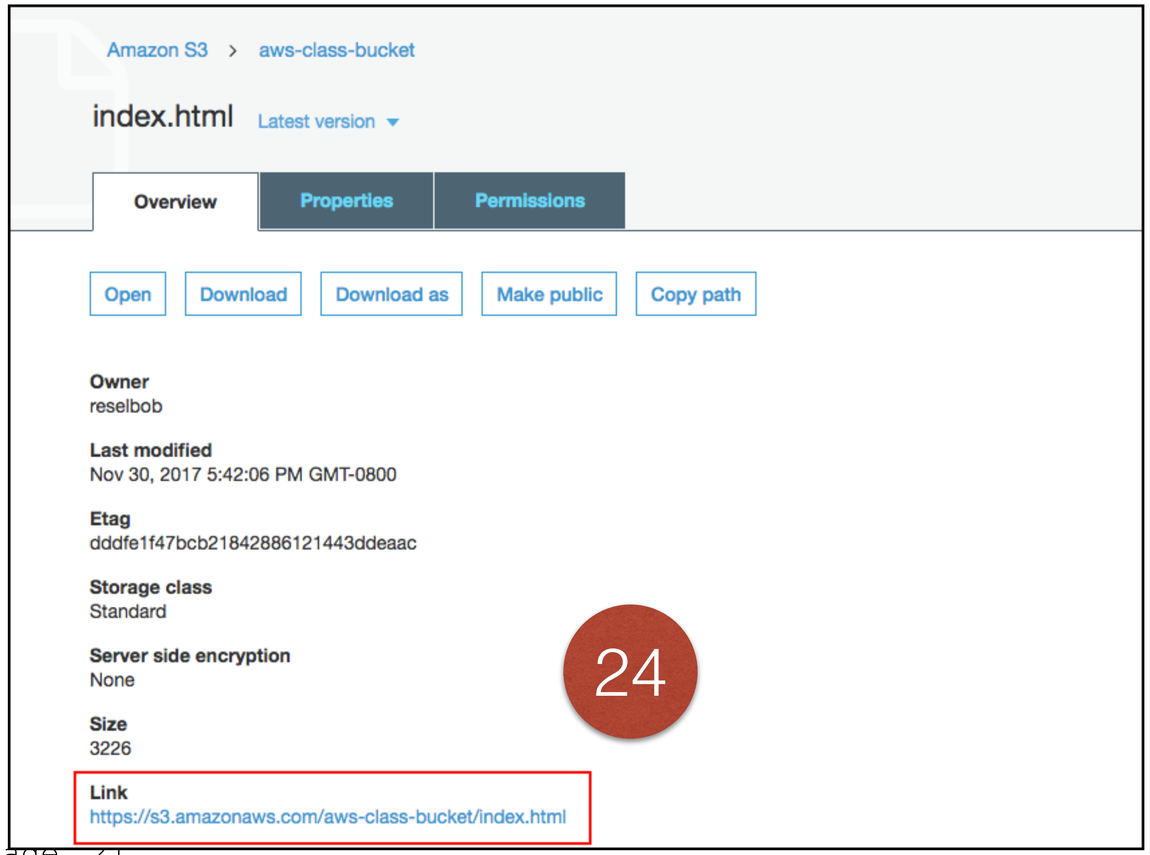
Create the S3 Bucket





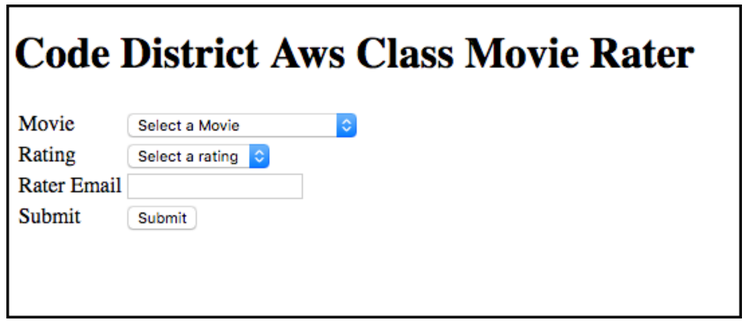
Create the S3 Bucket

https://s3.amazonaws.com/aws-class-bucket/index.html

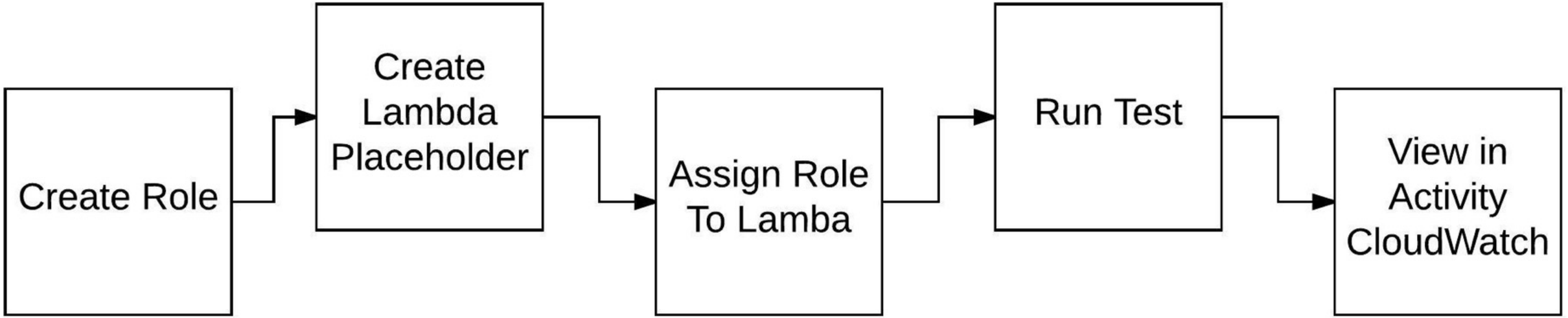


24

25

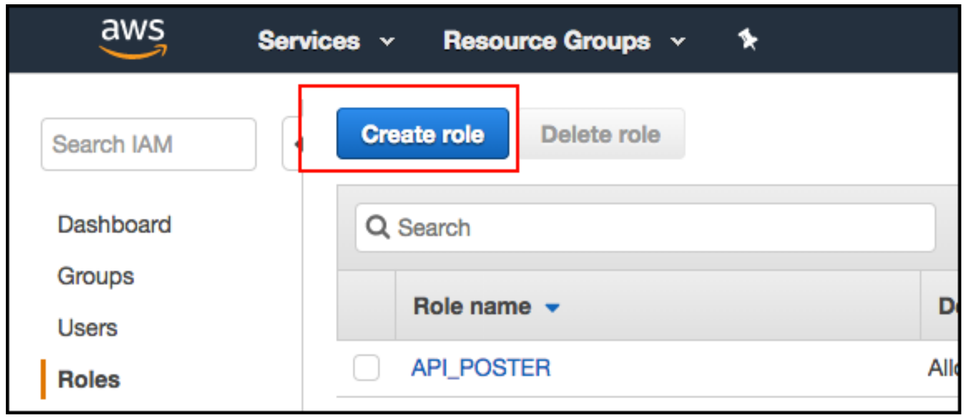


Session 2

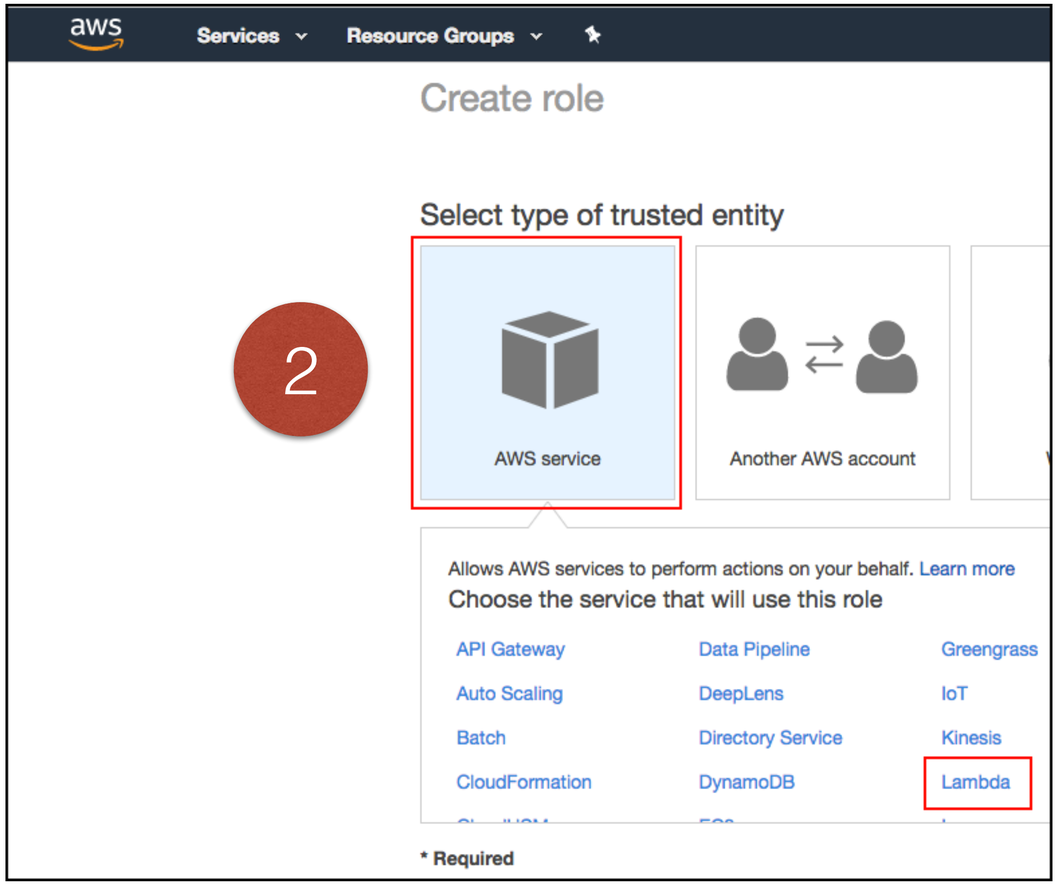

Create Role
Create Lambda Placeholder
Assign Role To Lamba
Run Test
View in Activity CloudWatch

Creating the Resource Role

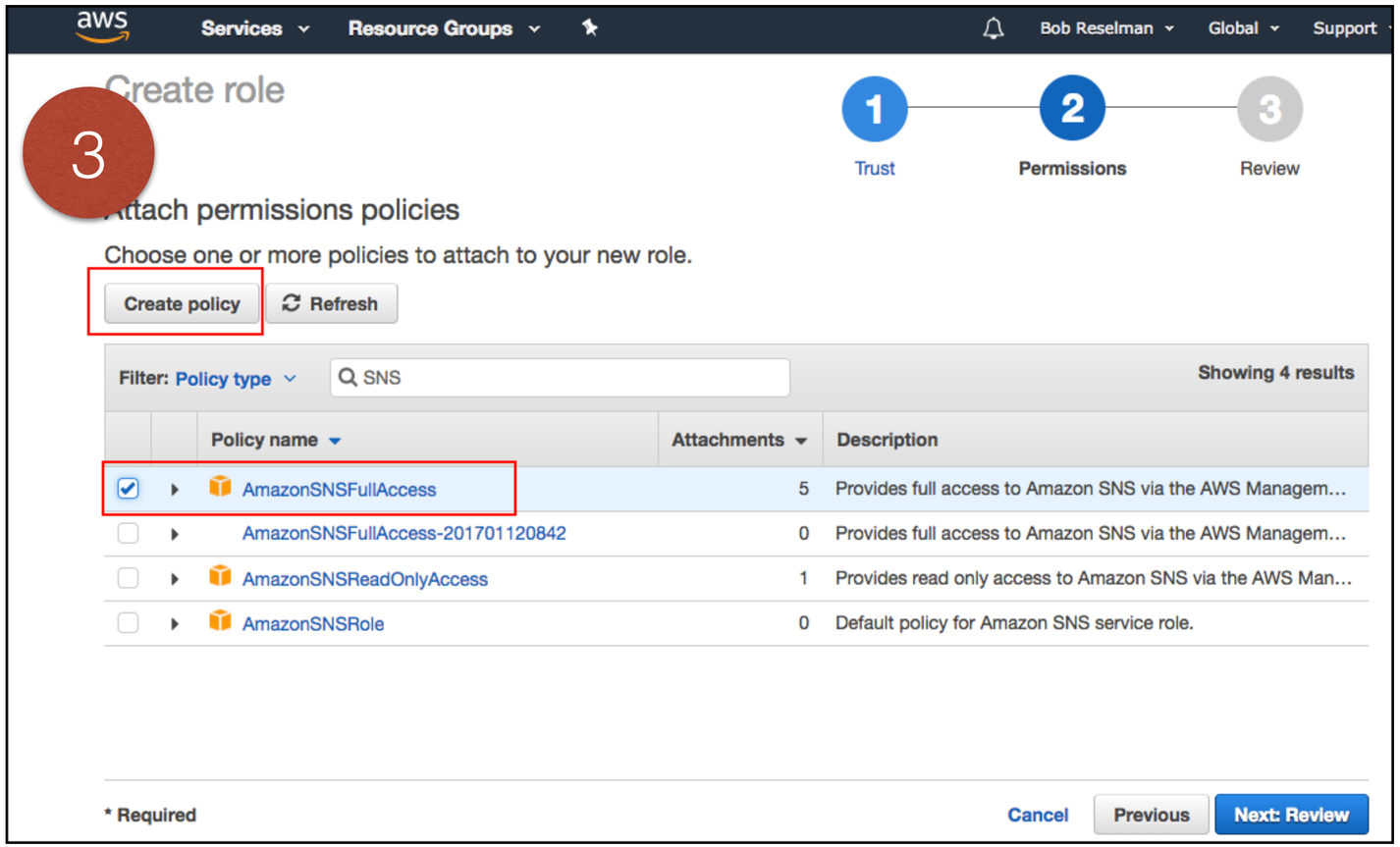
1

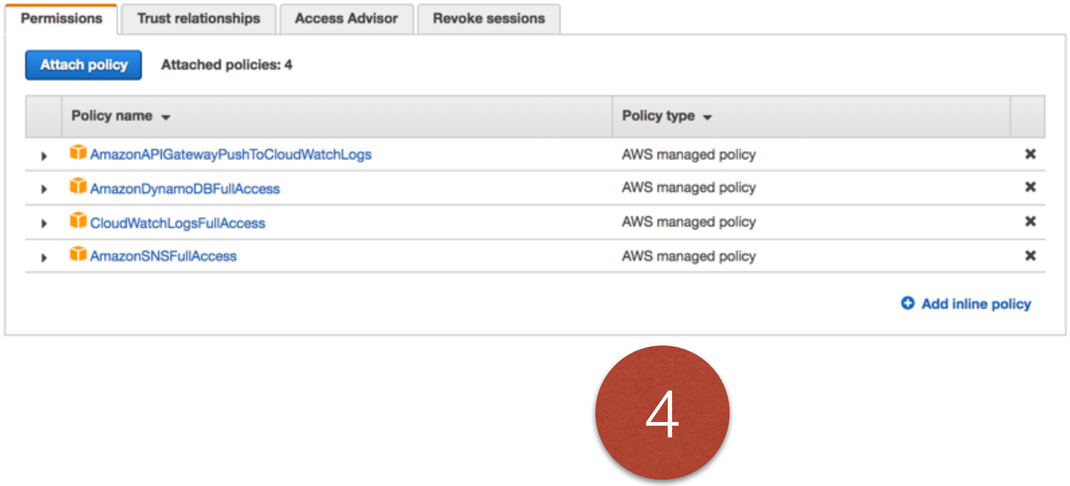


2



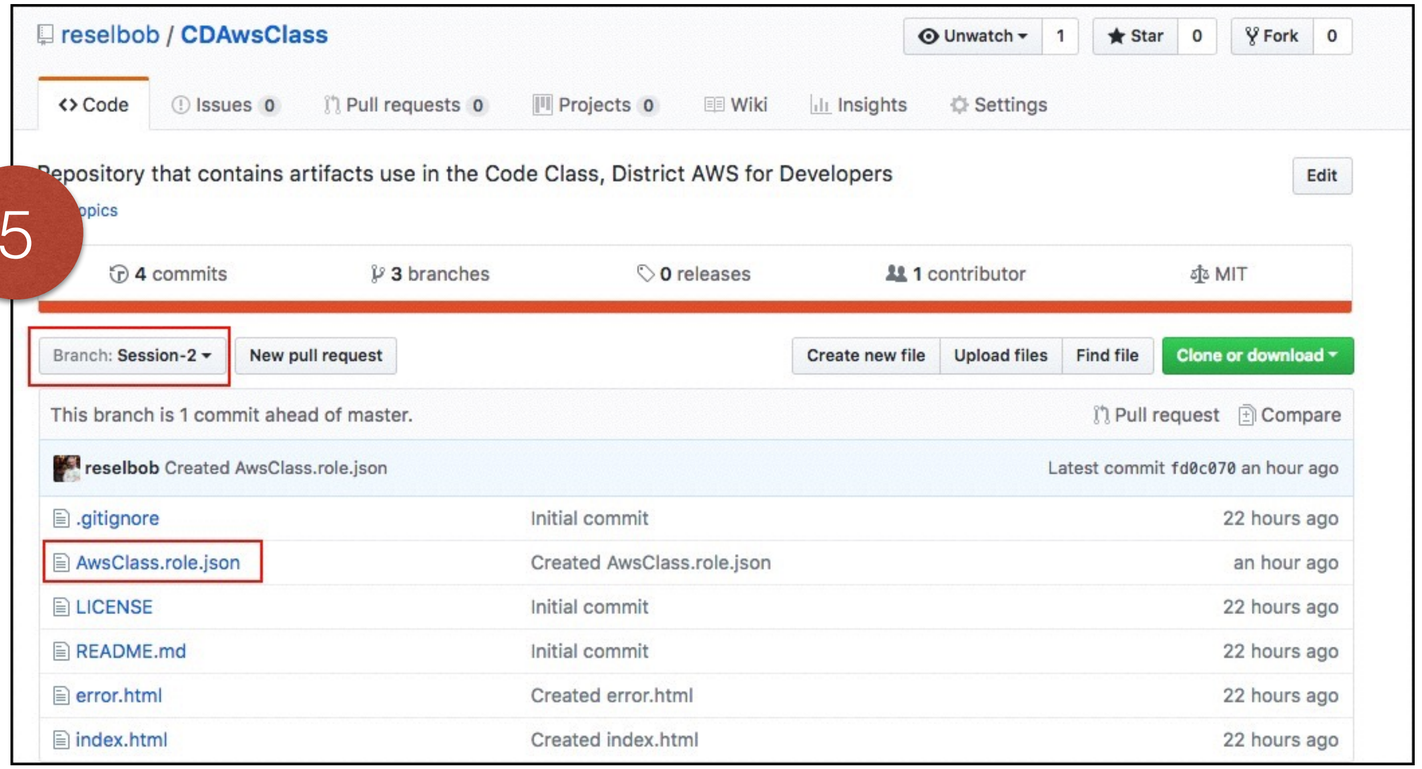
Creating the Resource Role





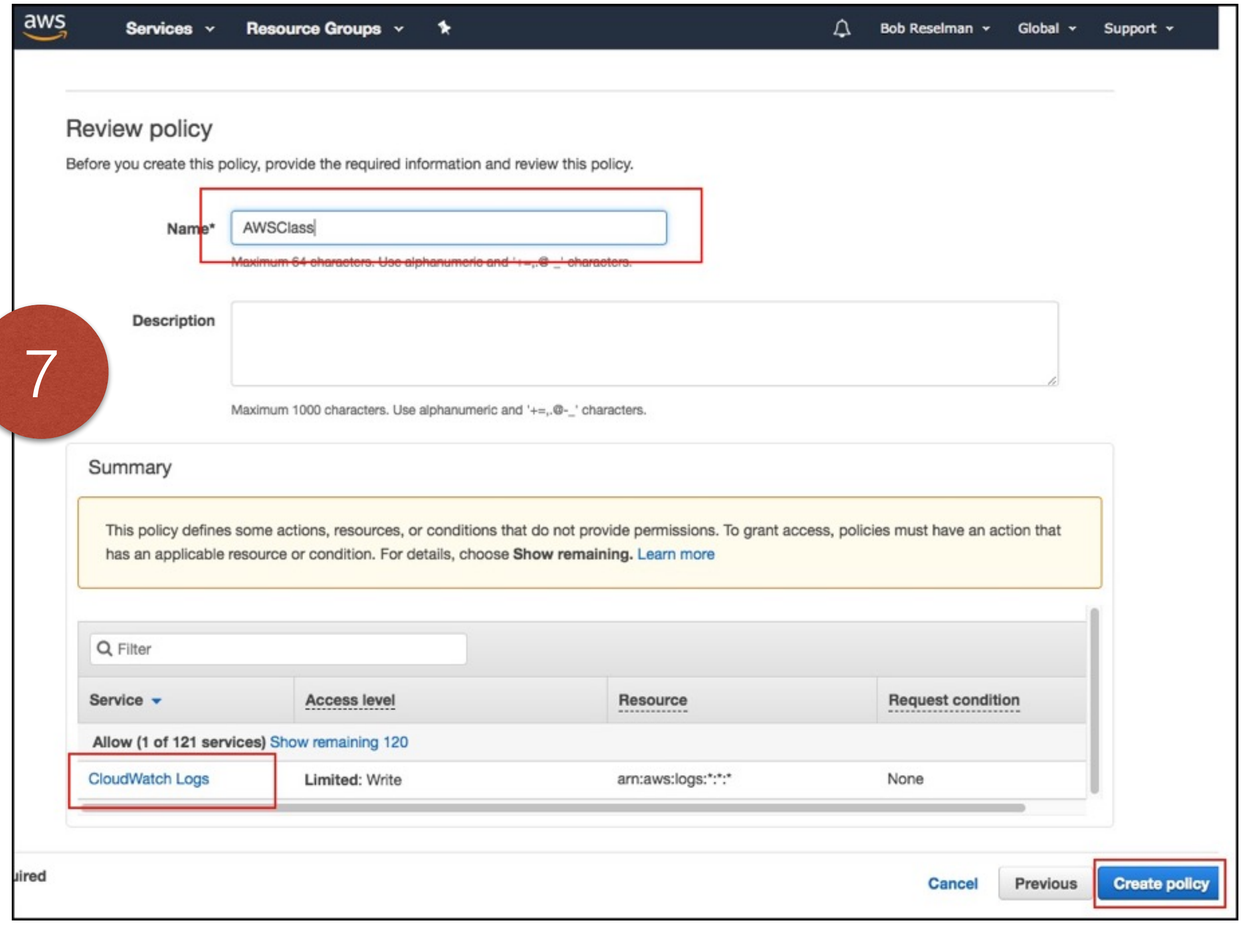
Creating the Resource Role

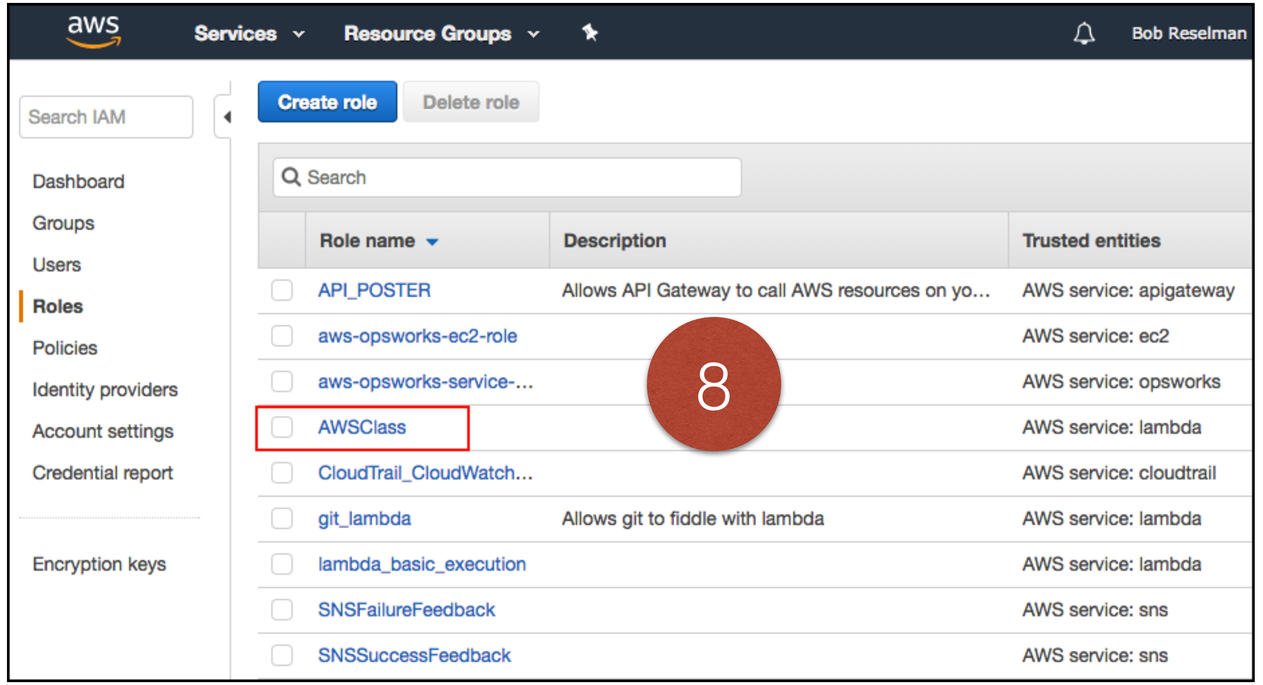
5



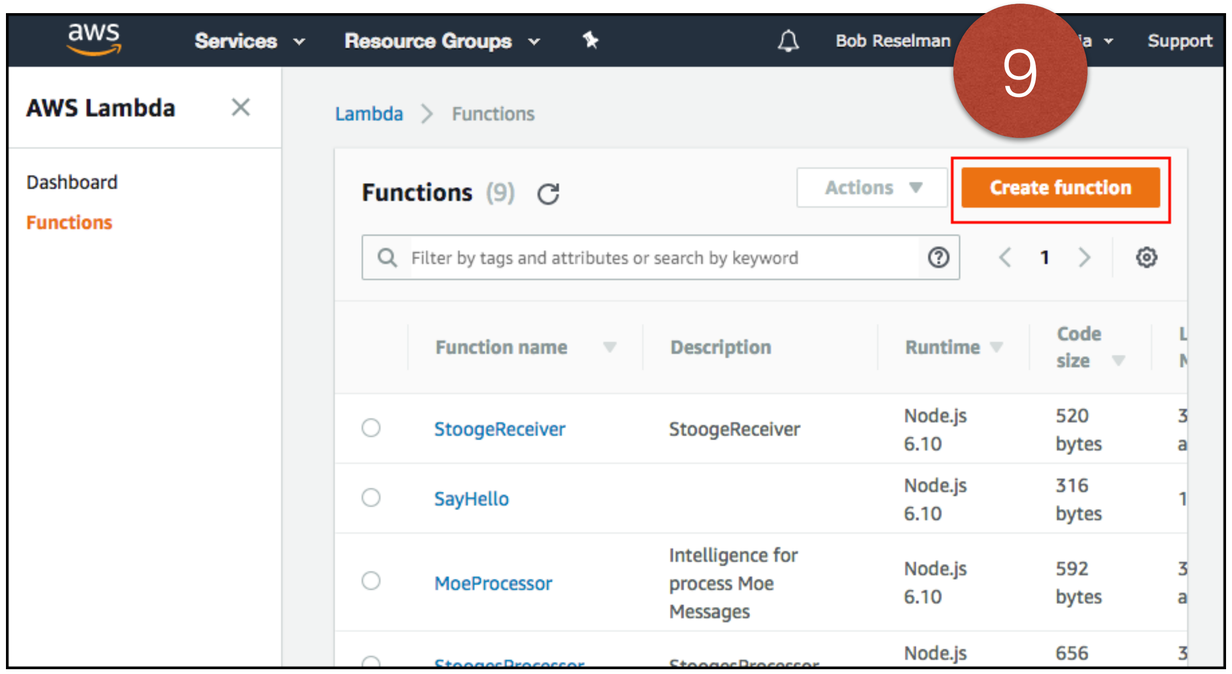
6

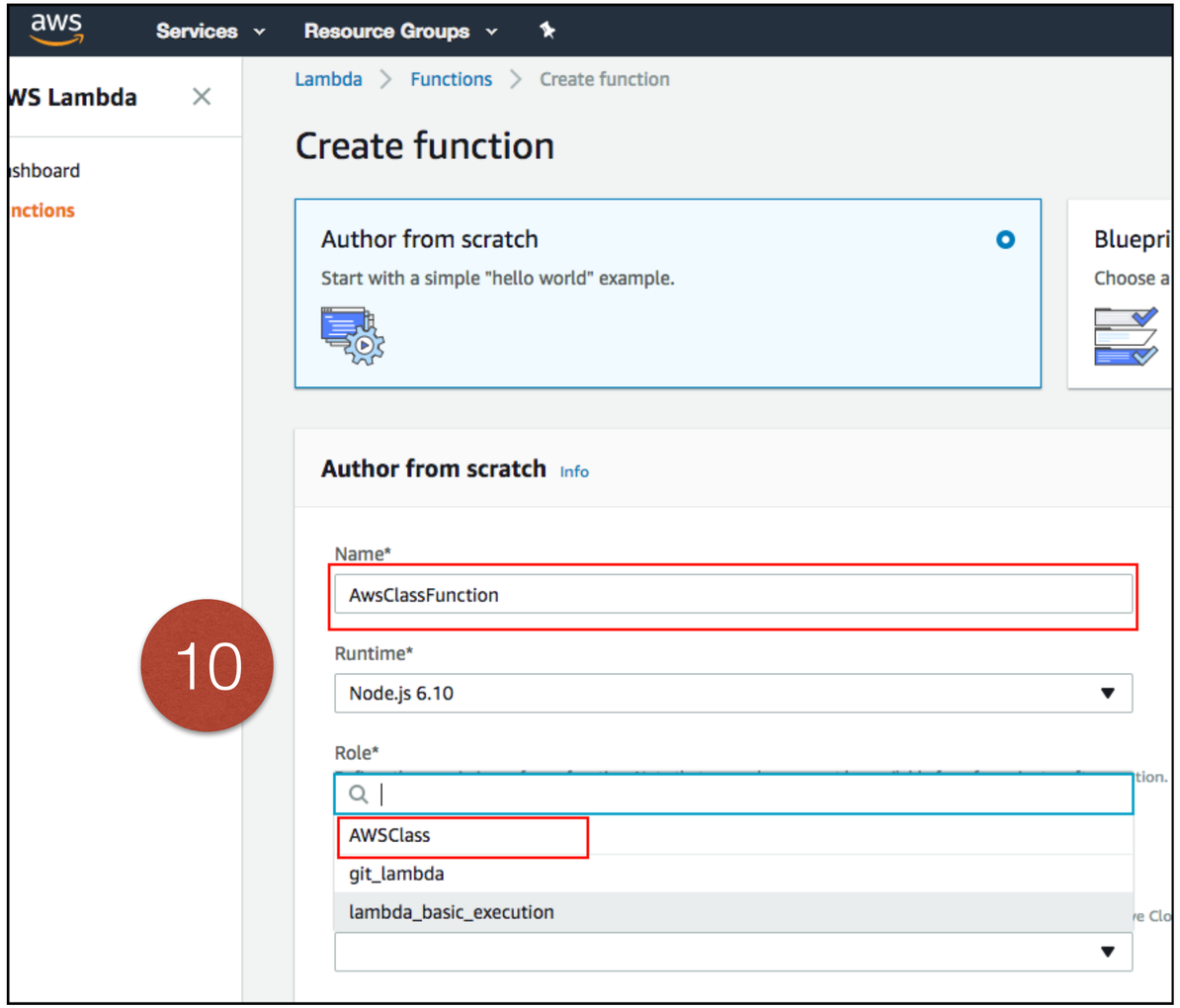
Creating the Resource Role



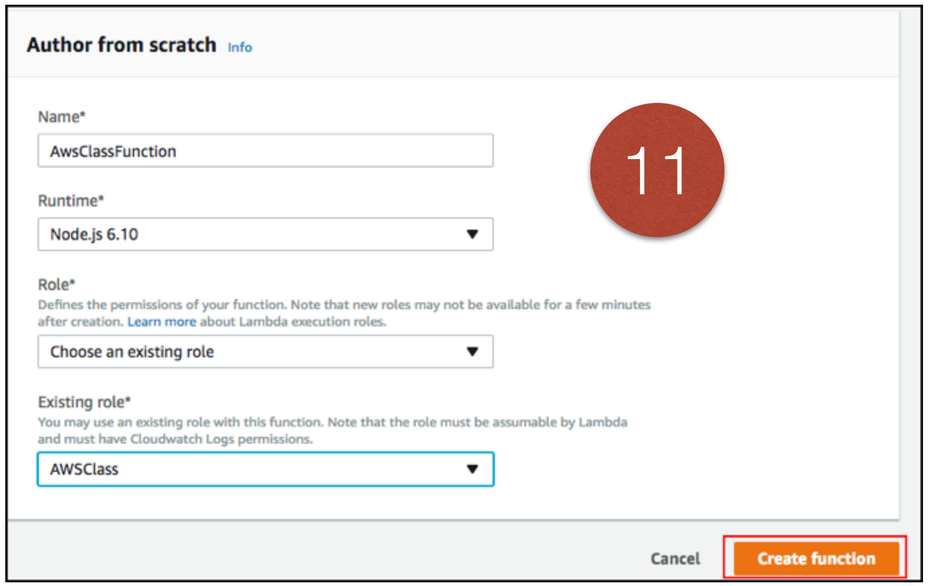


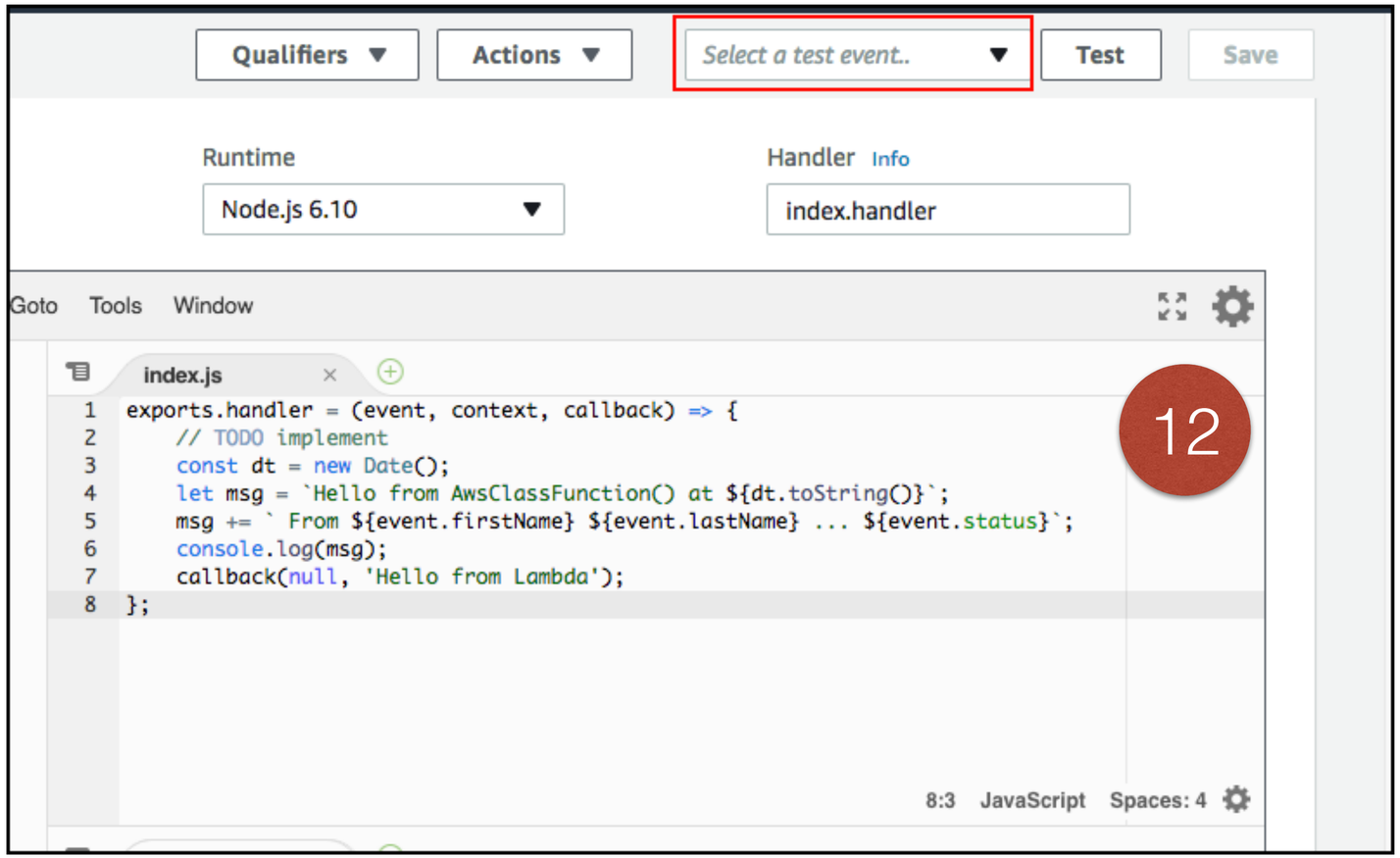
Creating Lambda Function



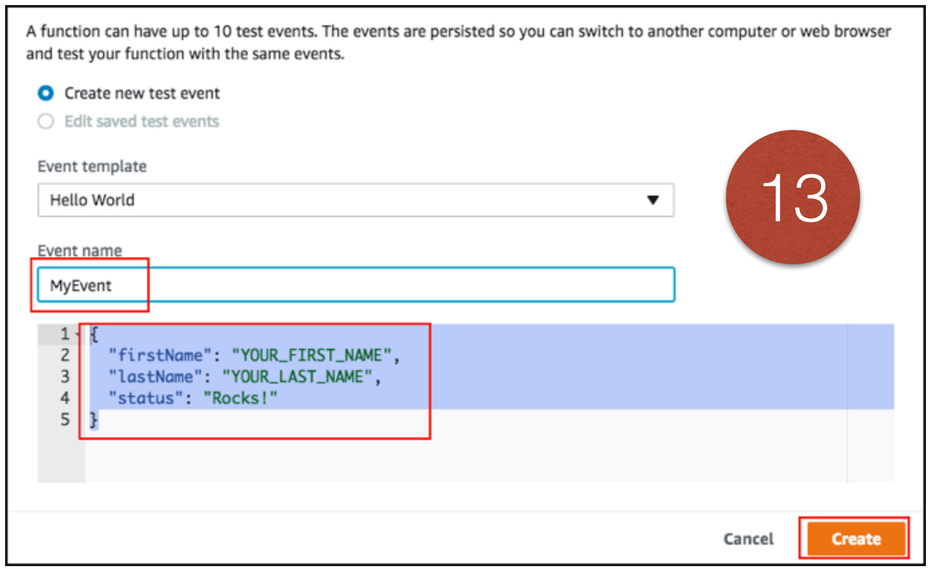


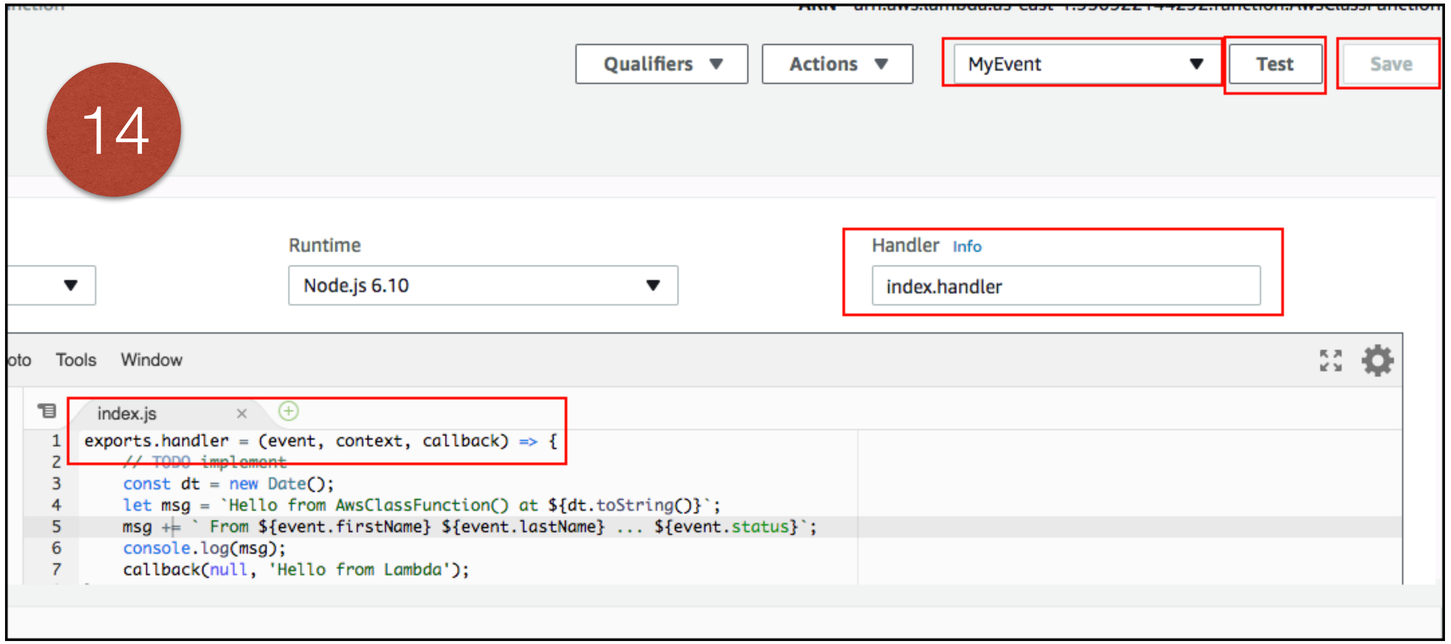
Creating Lambda Function



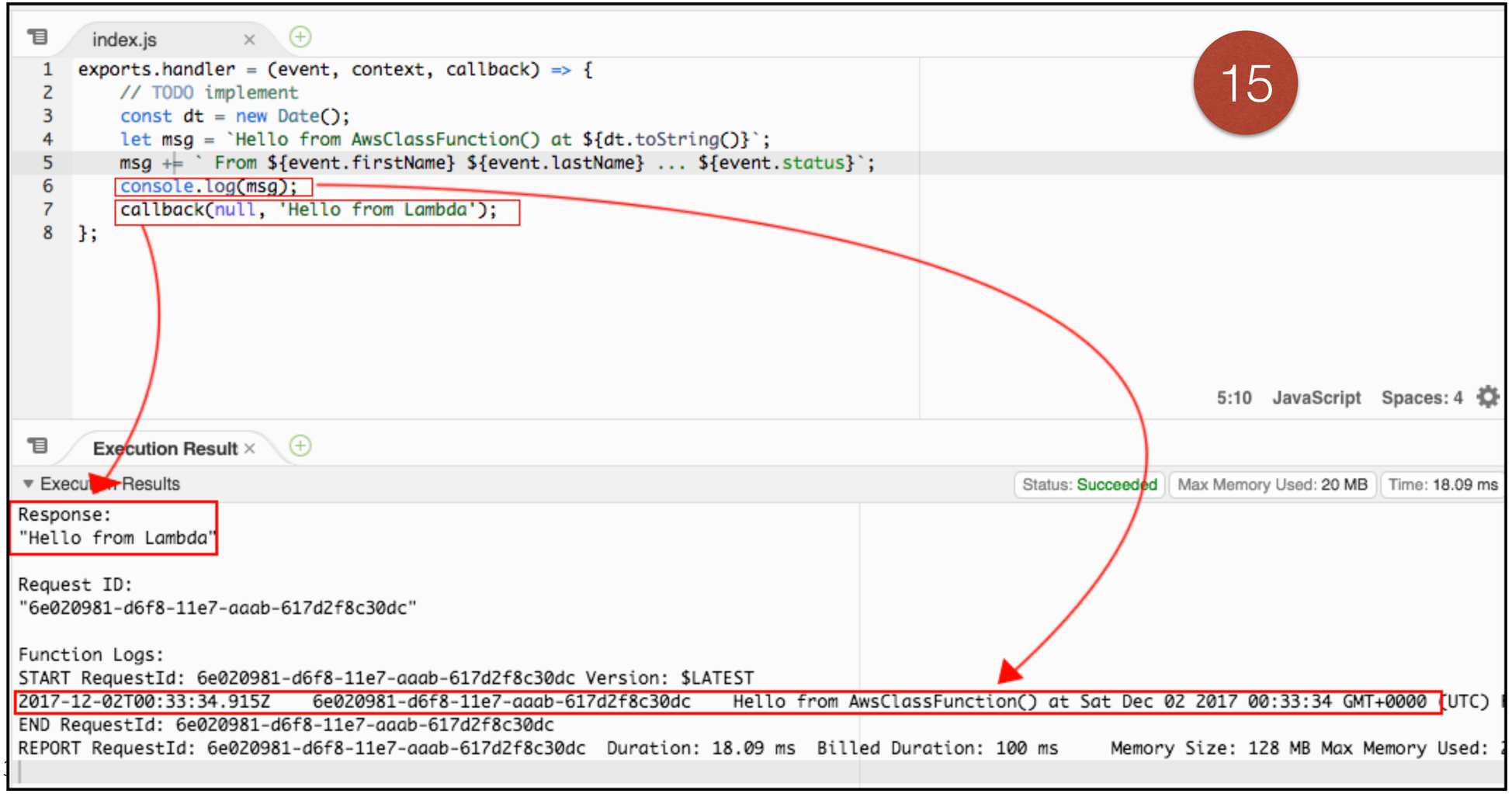


Creating Lambda Function

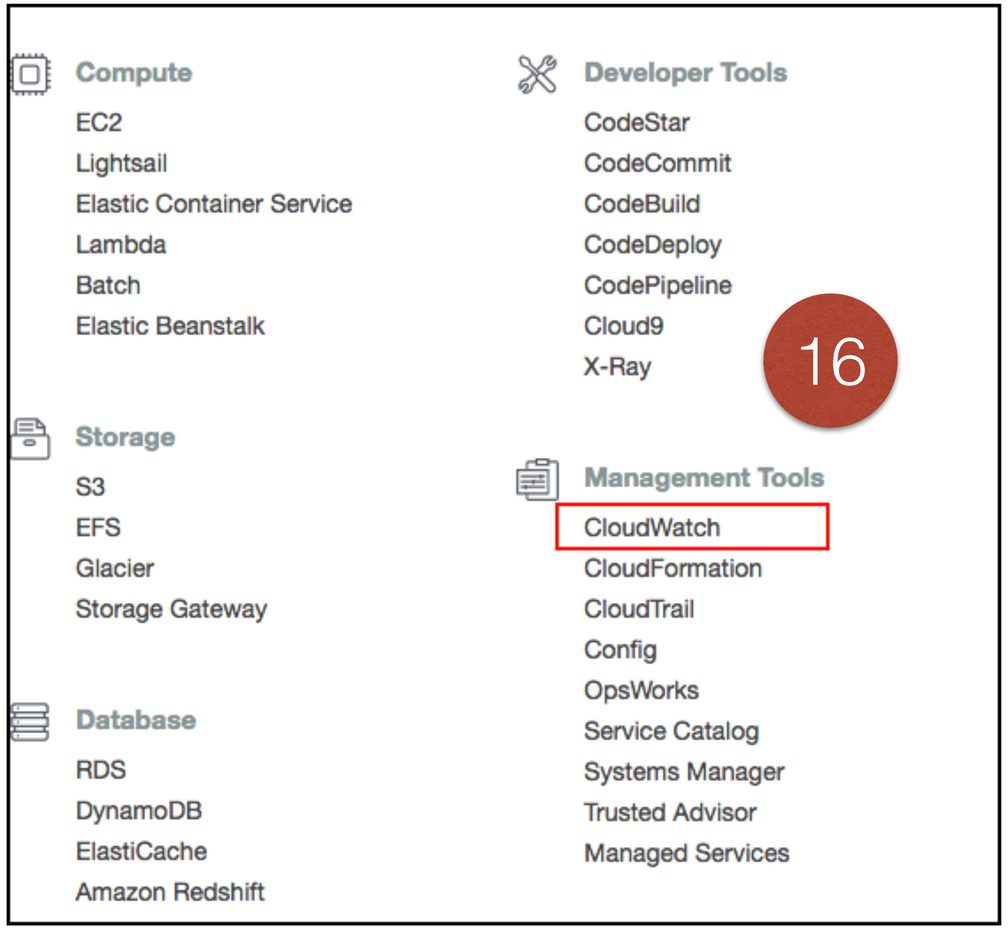


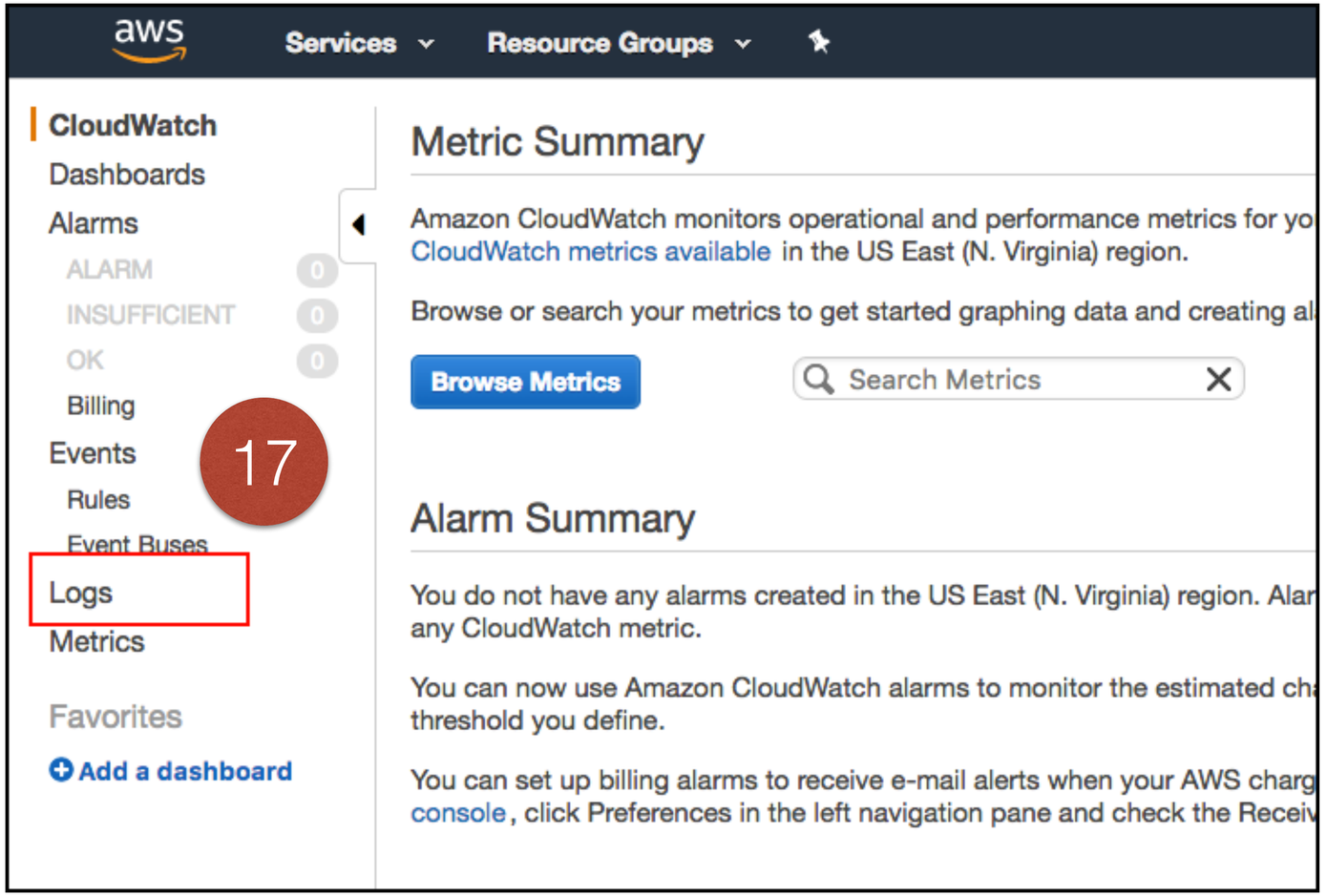


Creating a Lambda Function

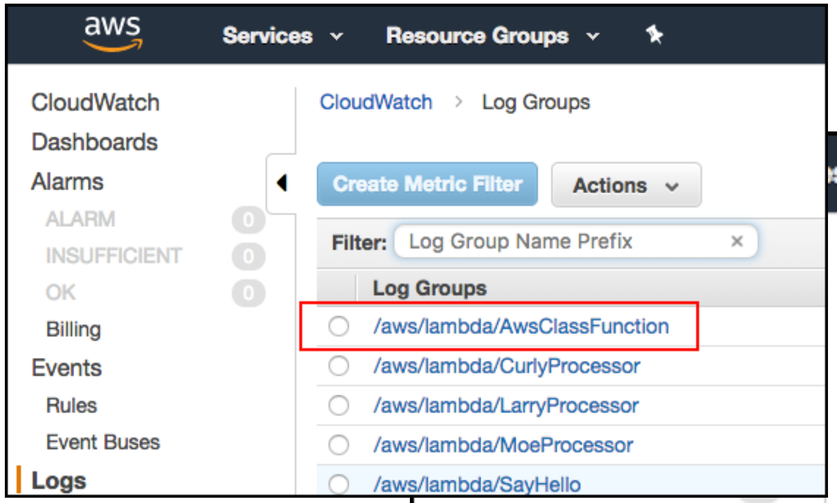


View Activity in CloudWatch

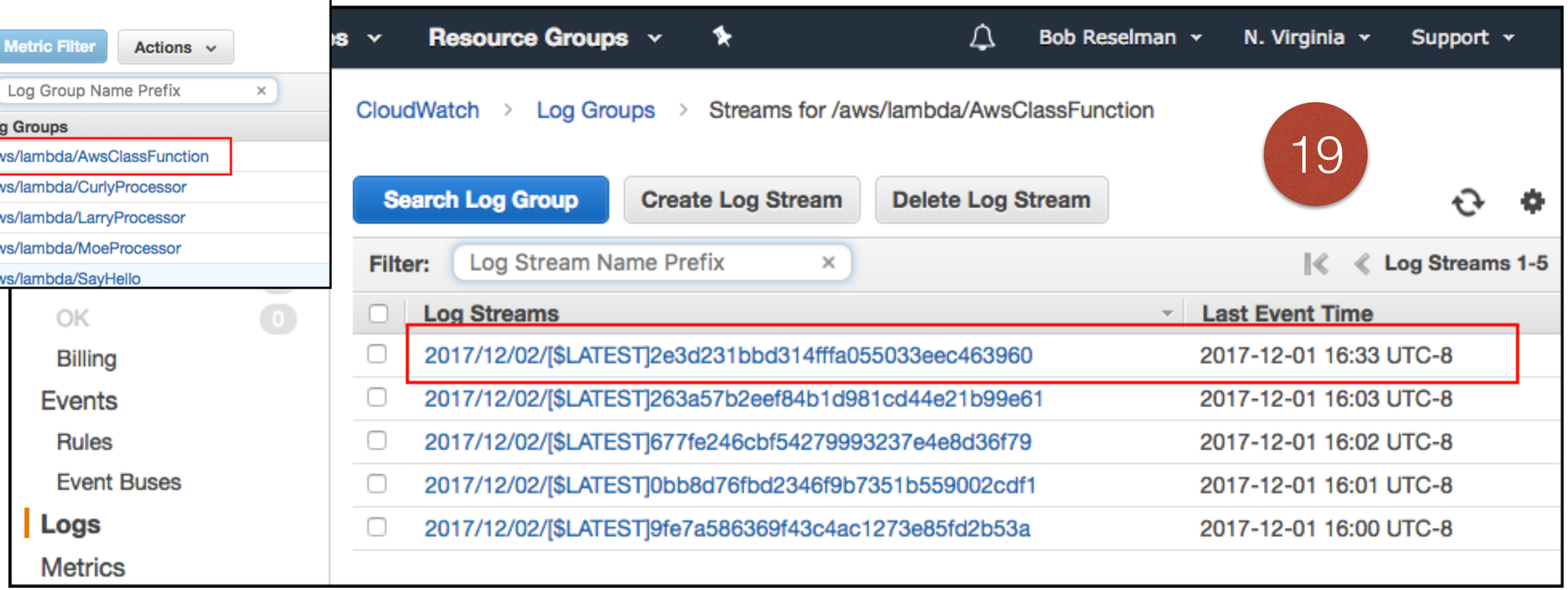




View Activity in CloudWatch

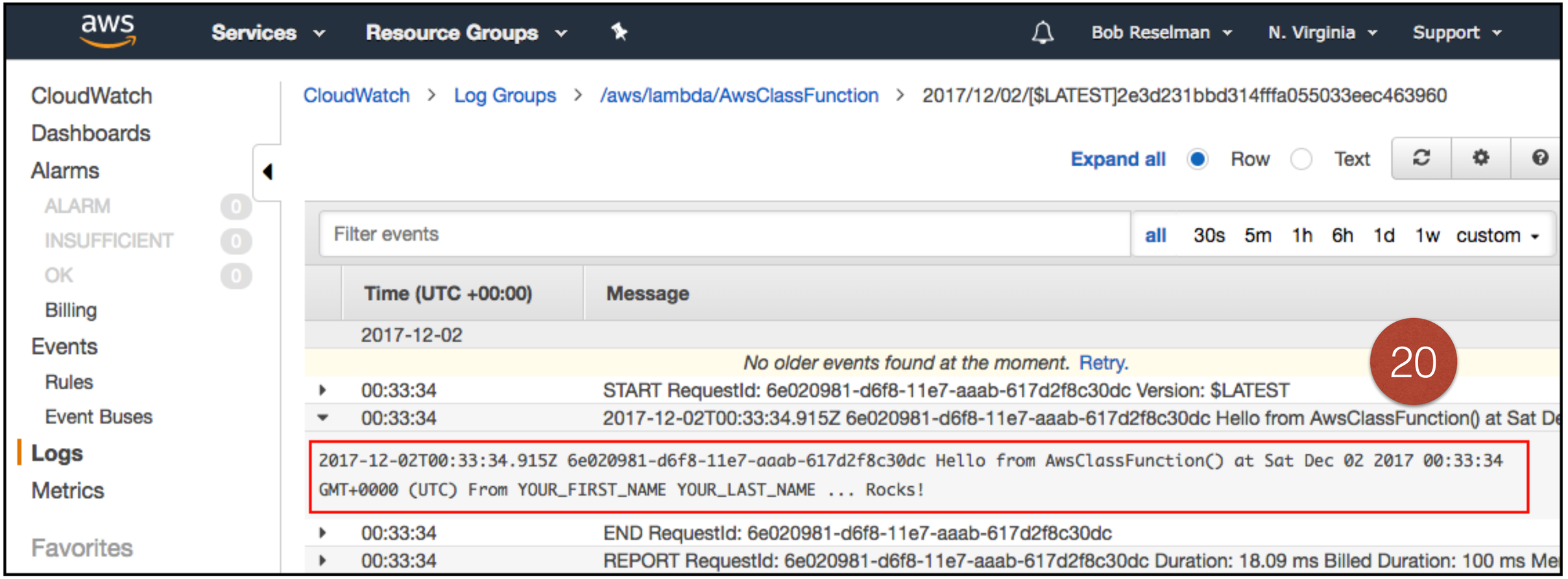


18

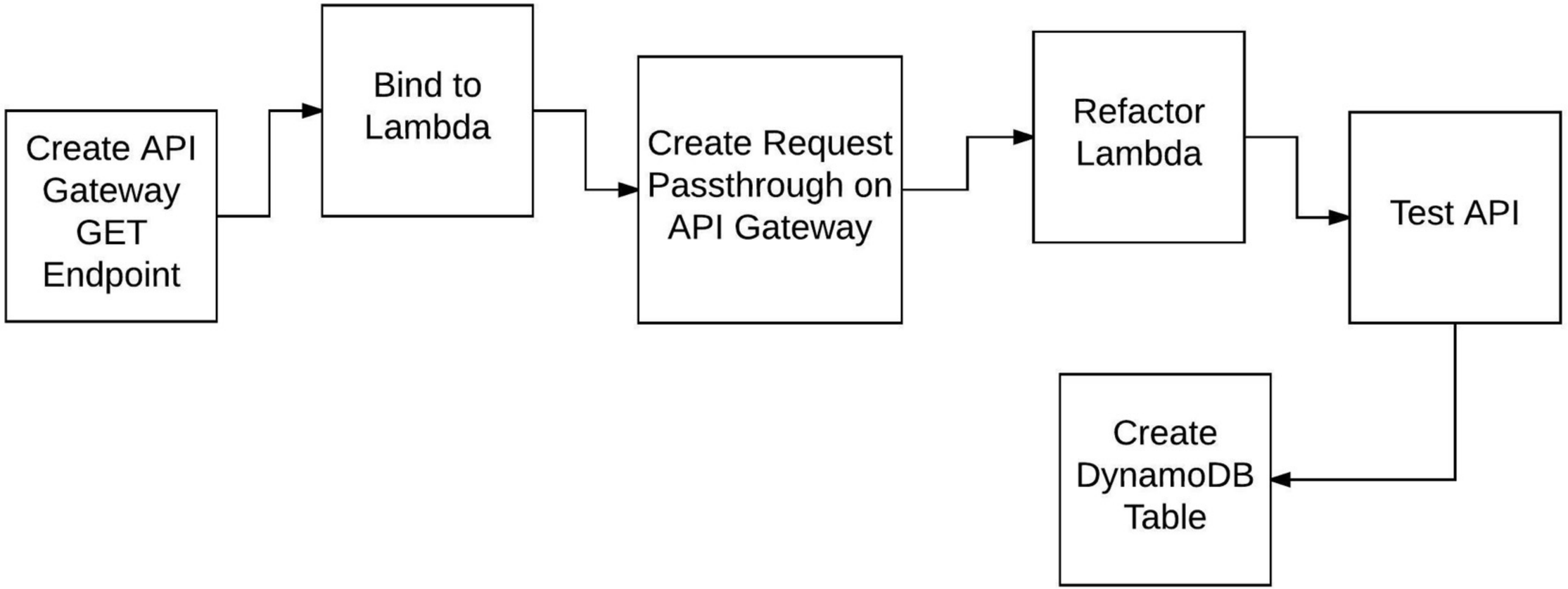


19

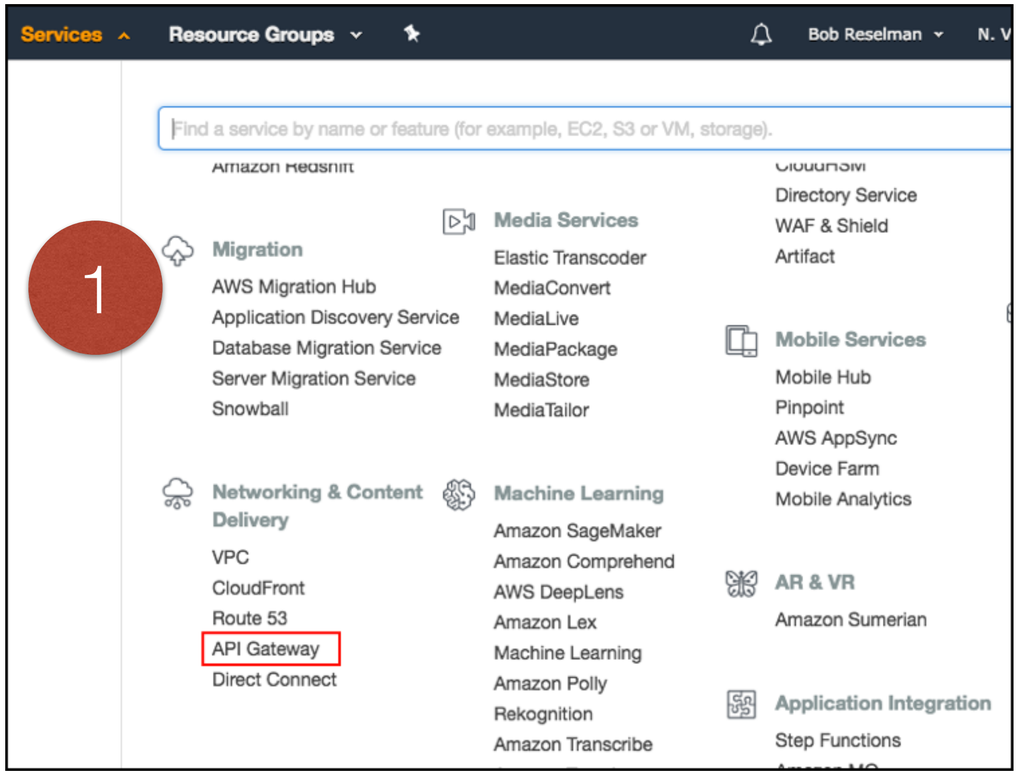
View Activity in CloudWatch

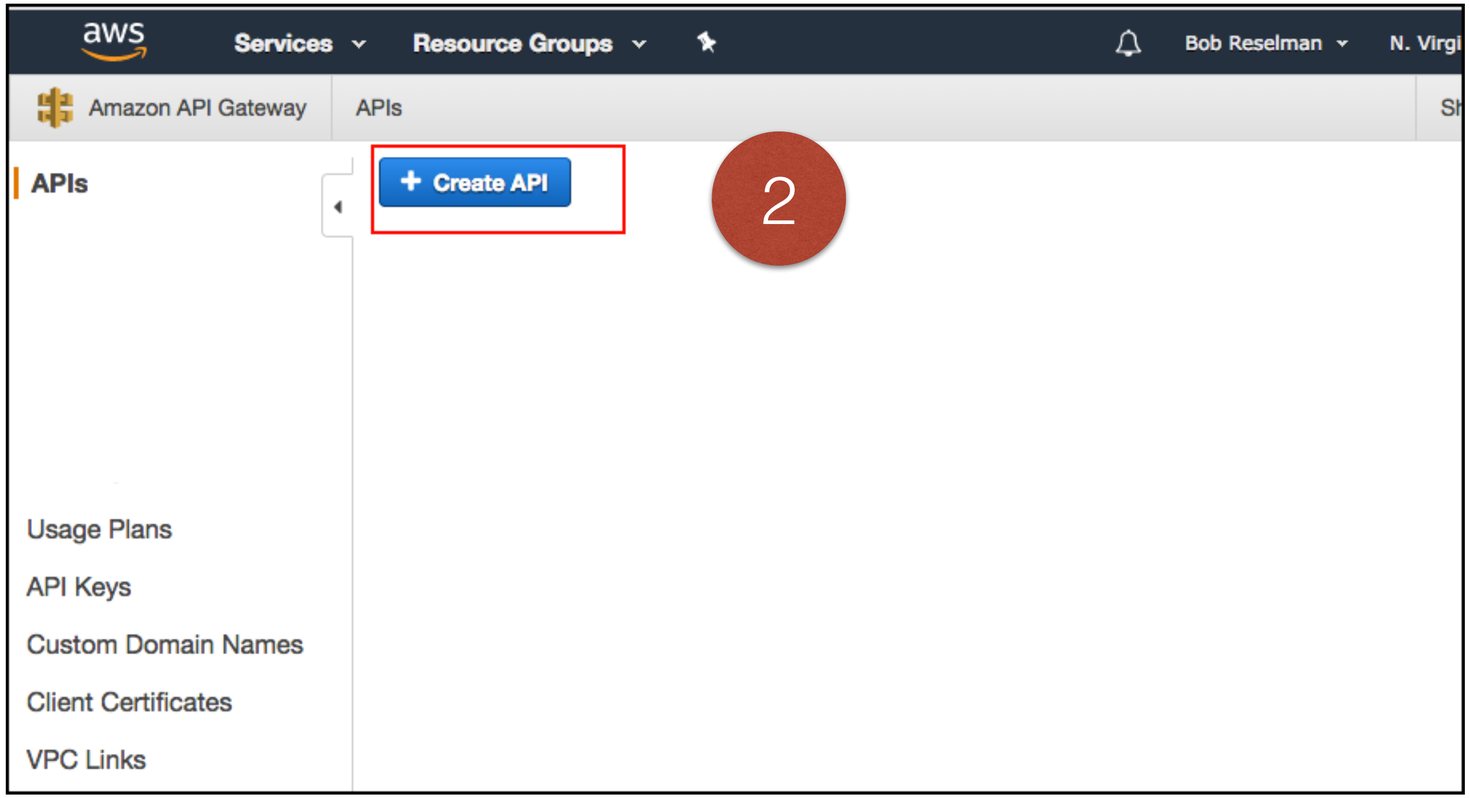


Session 3

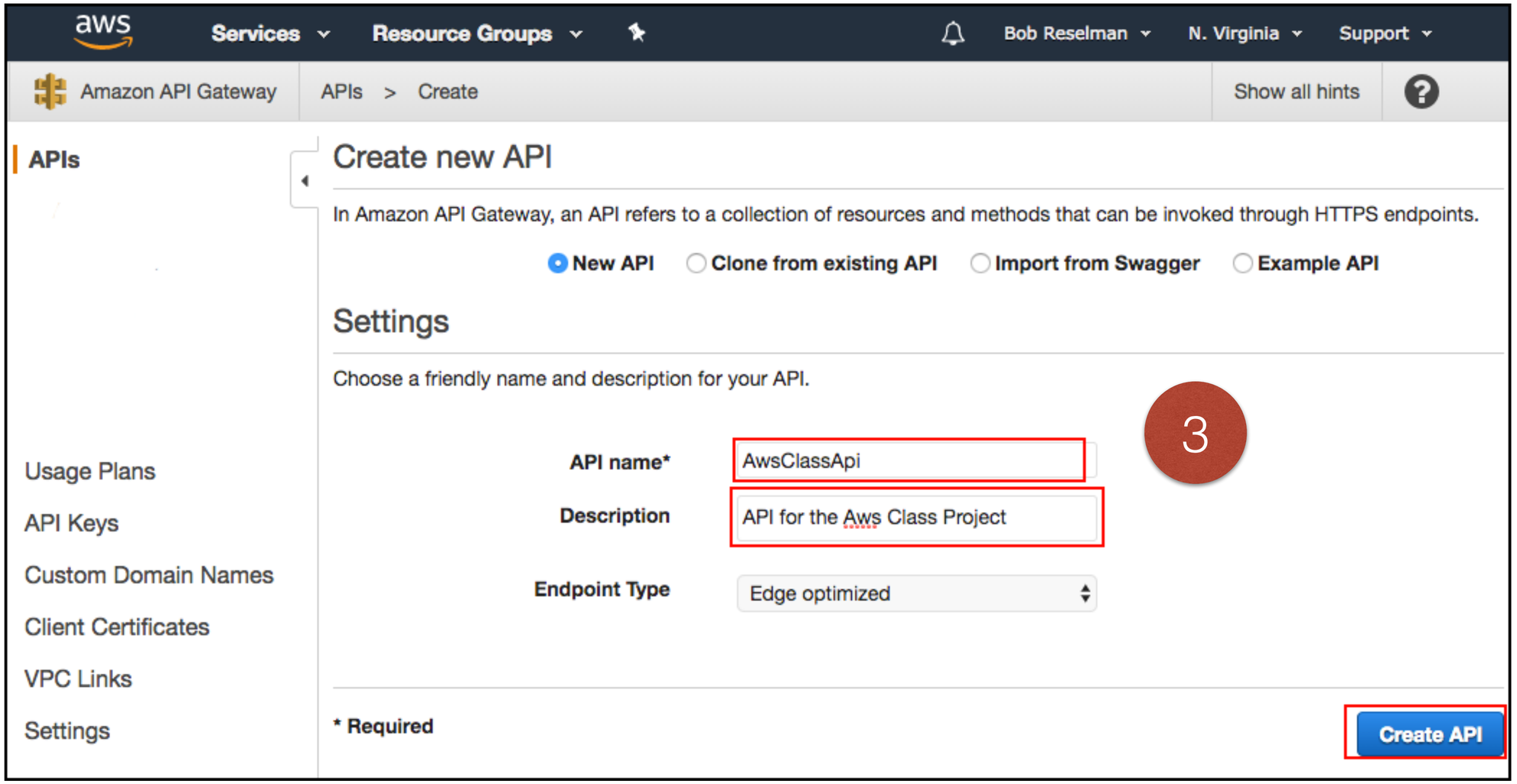

Create API Gateway GET Endpoint
Bind to Lambda
Create Request Passthrough on API Gateway
Refactor Lambda
Test API
Create DynamoDB Table

Creating API GET Endpoint

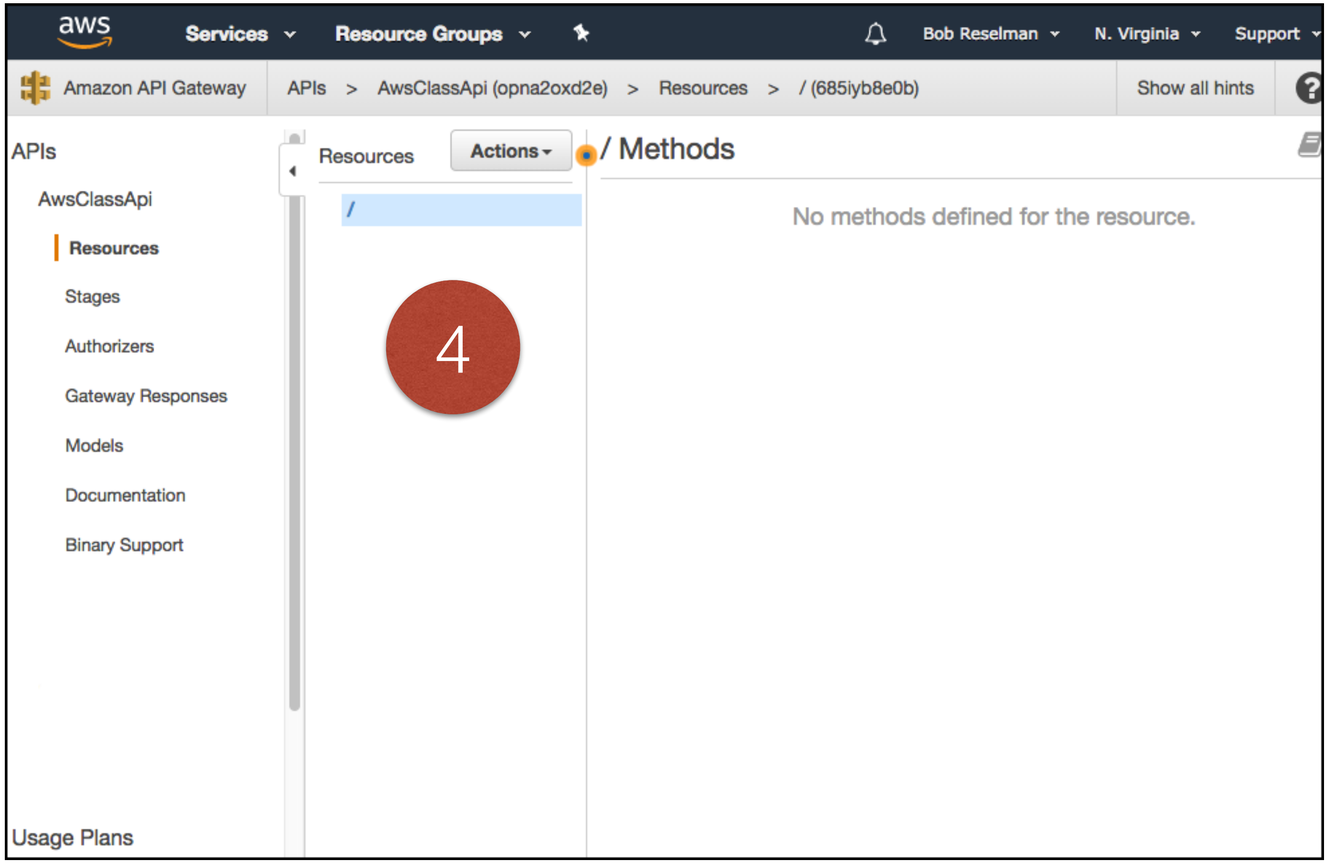


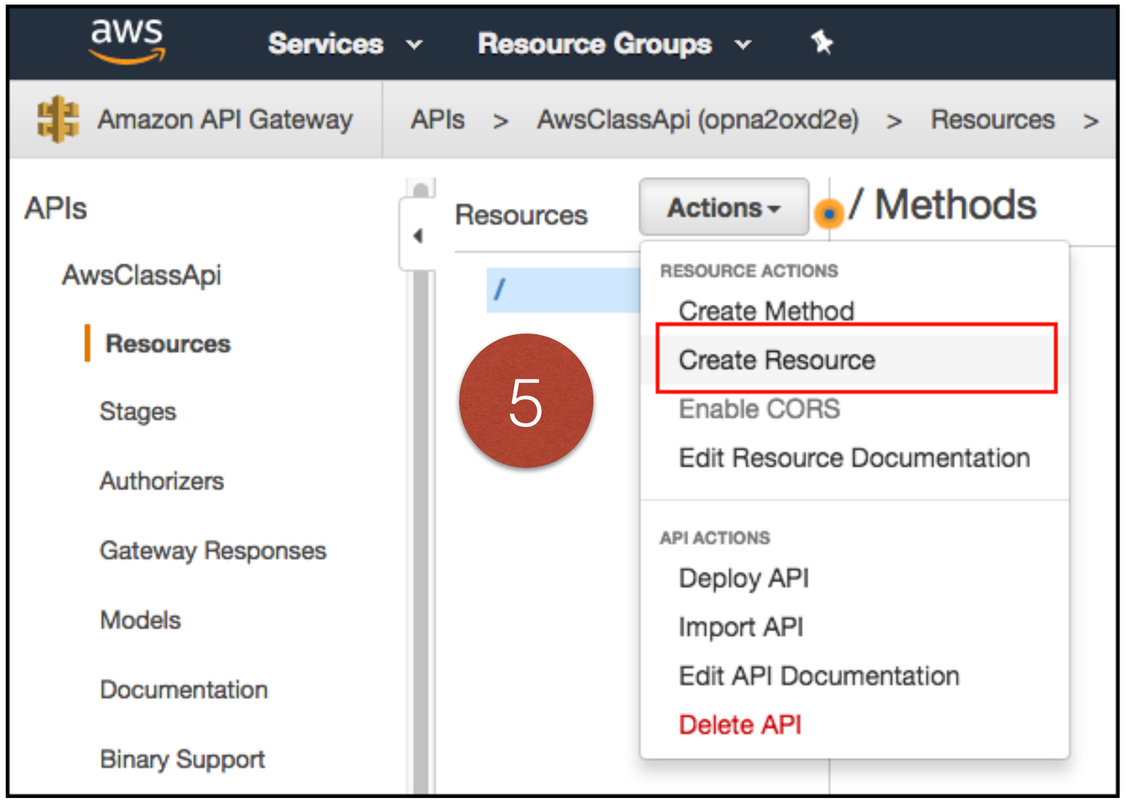


Creating API GET Endpoint

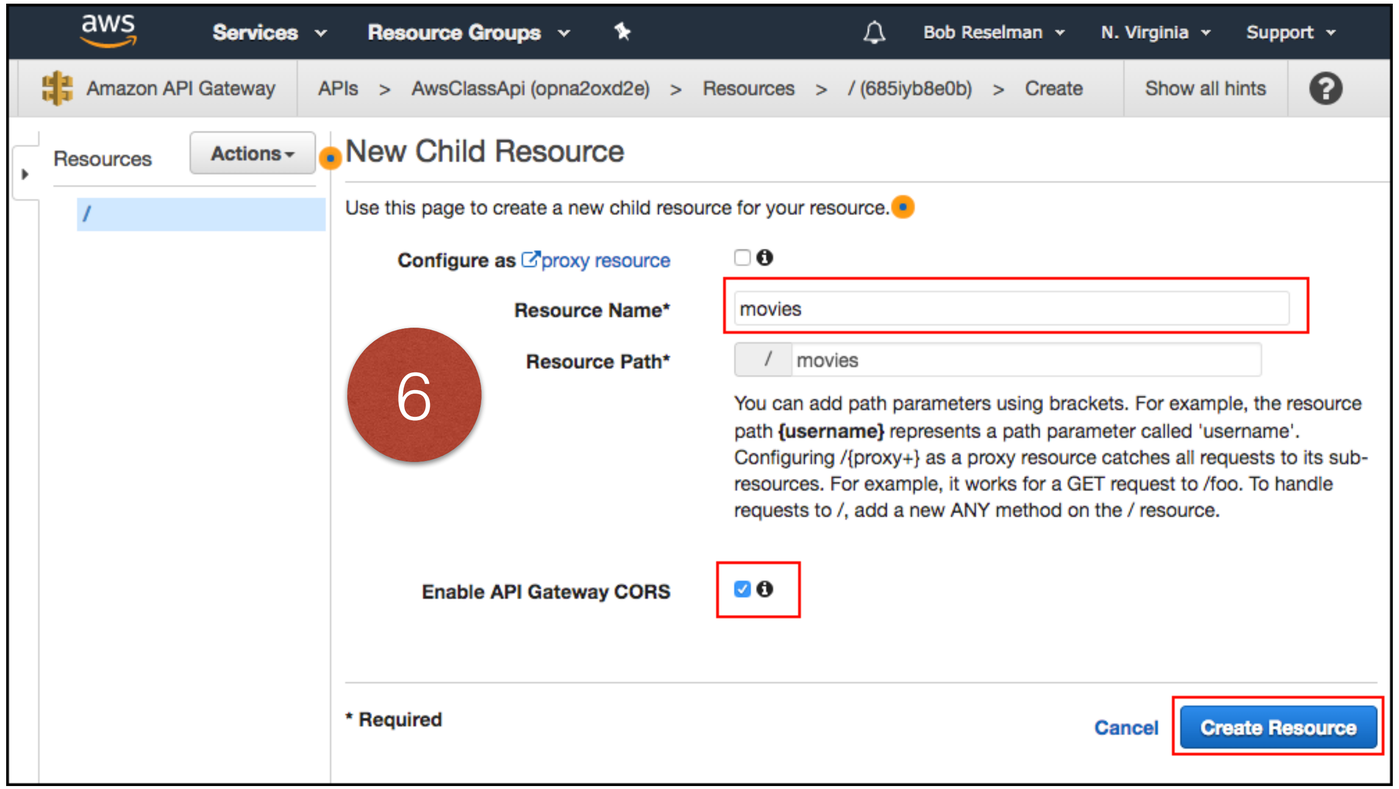


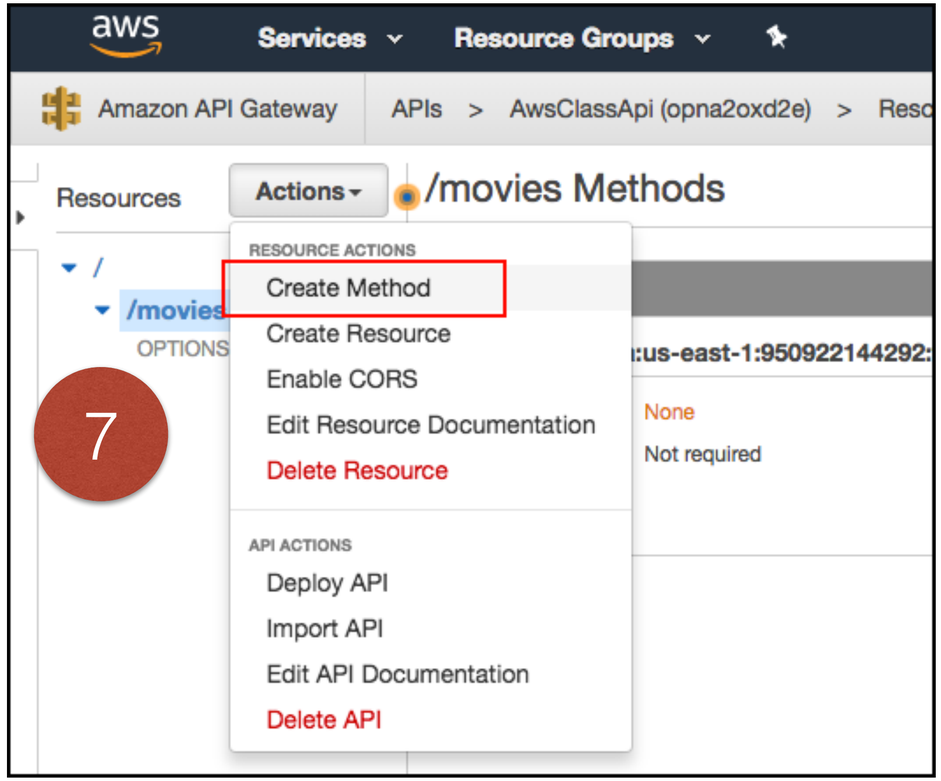
Creating API GET Endpoint



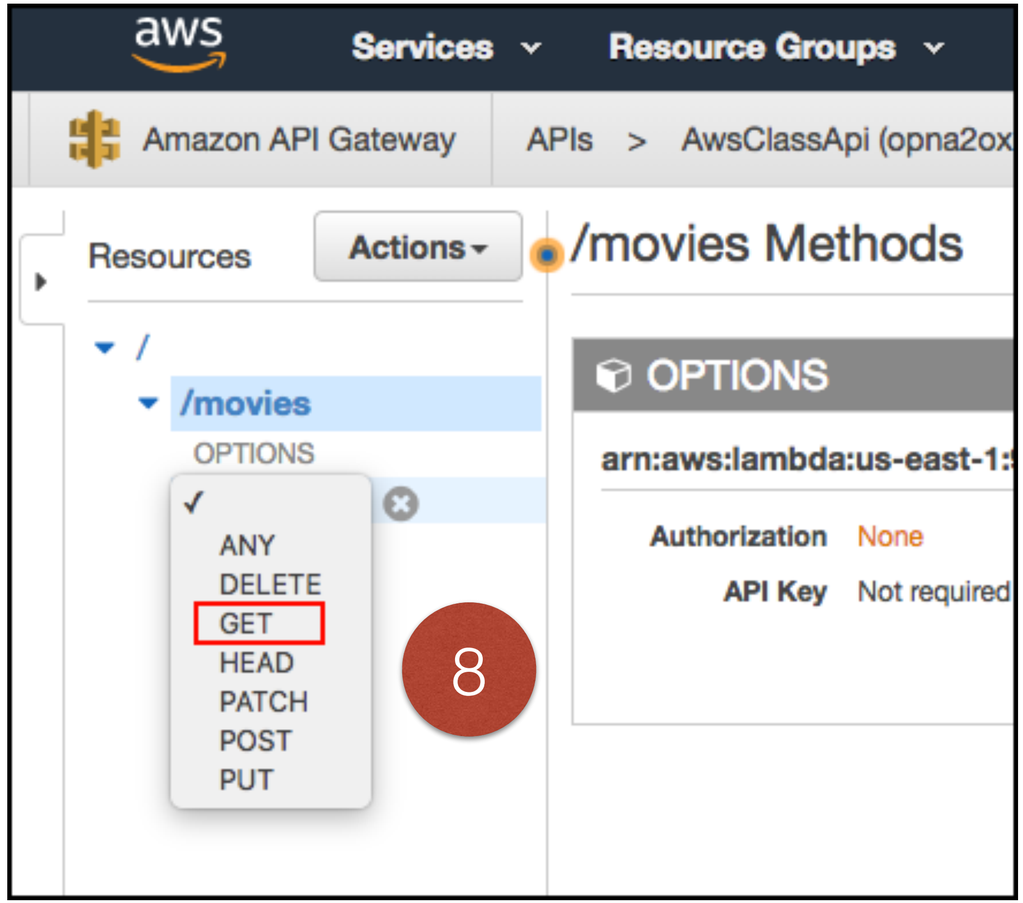


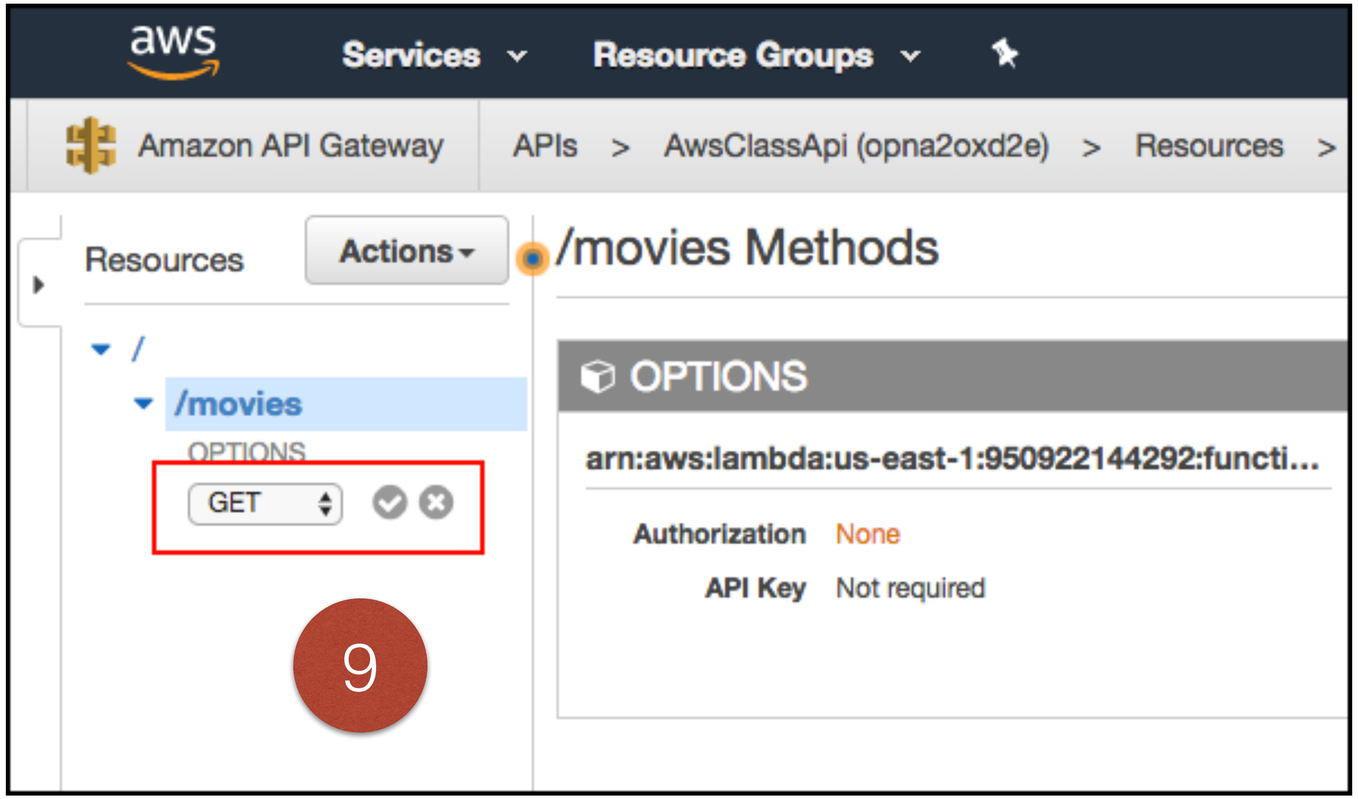
Creating API GET Endpoint



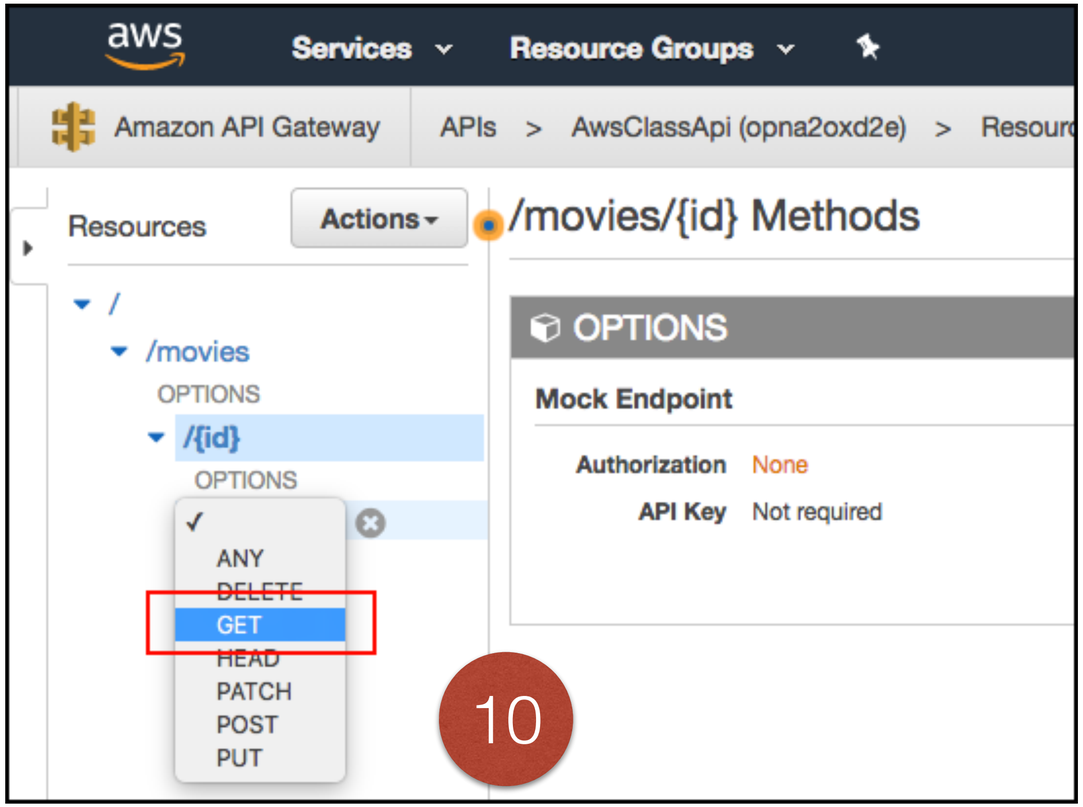


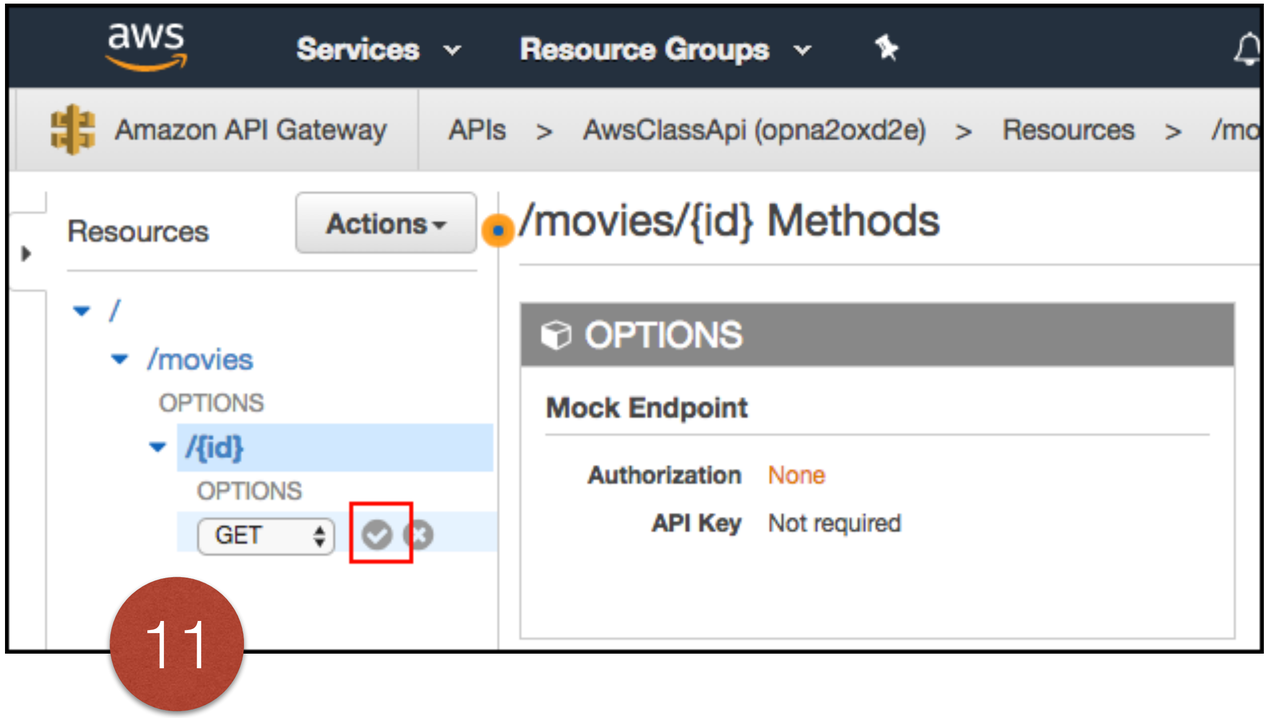
Creating API GET Endpoint



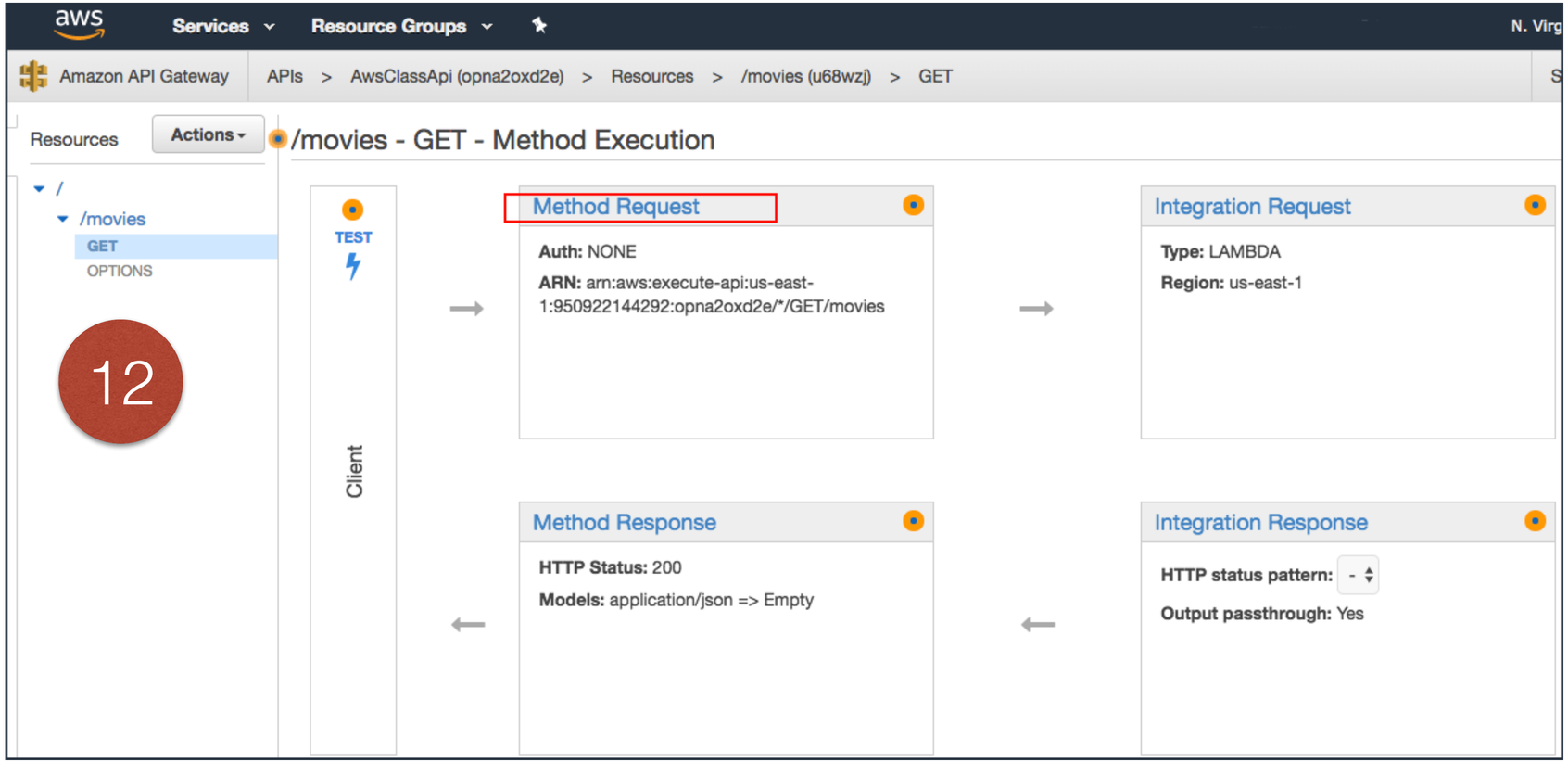


Creating API GET Endpoint

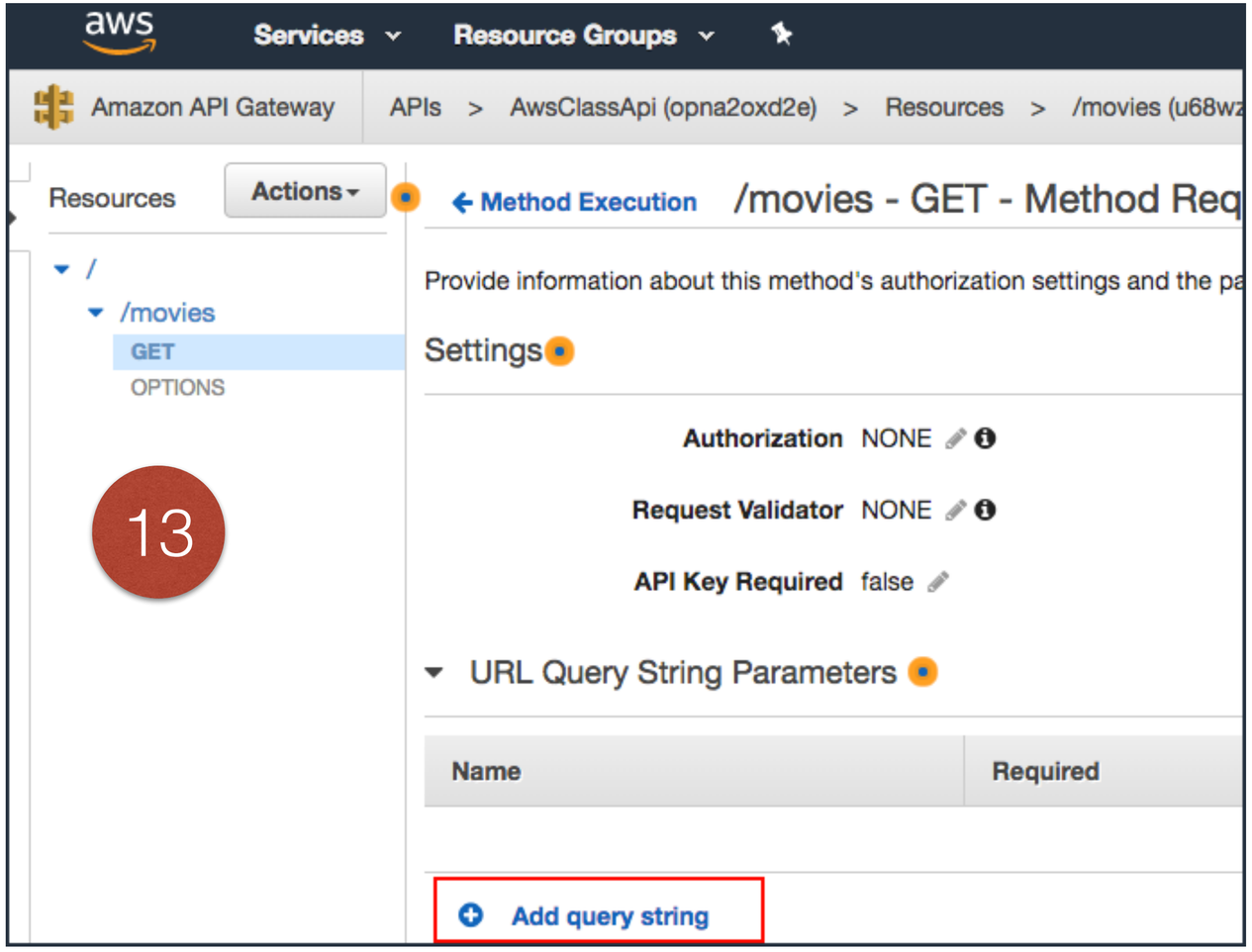


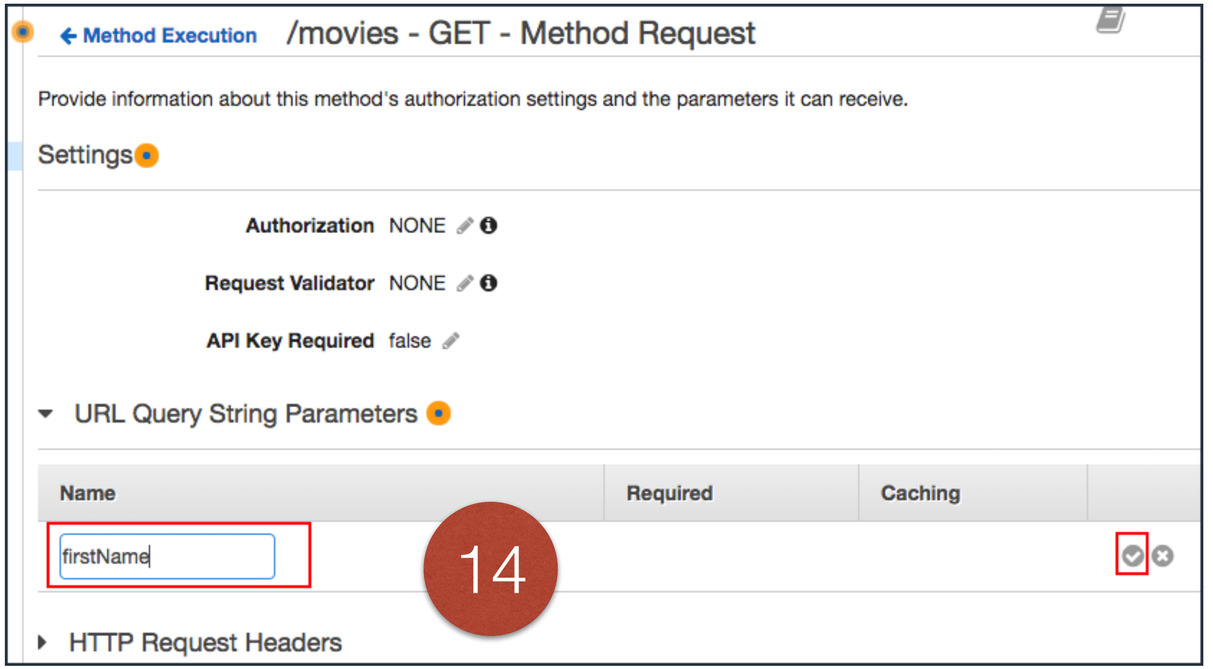


Creating API GET Endpoint

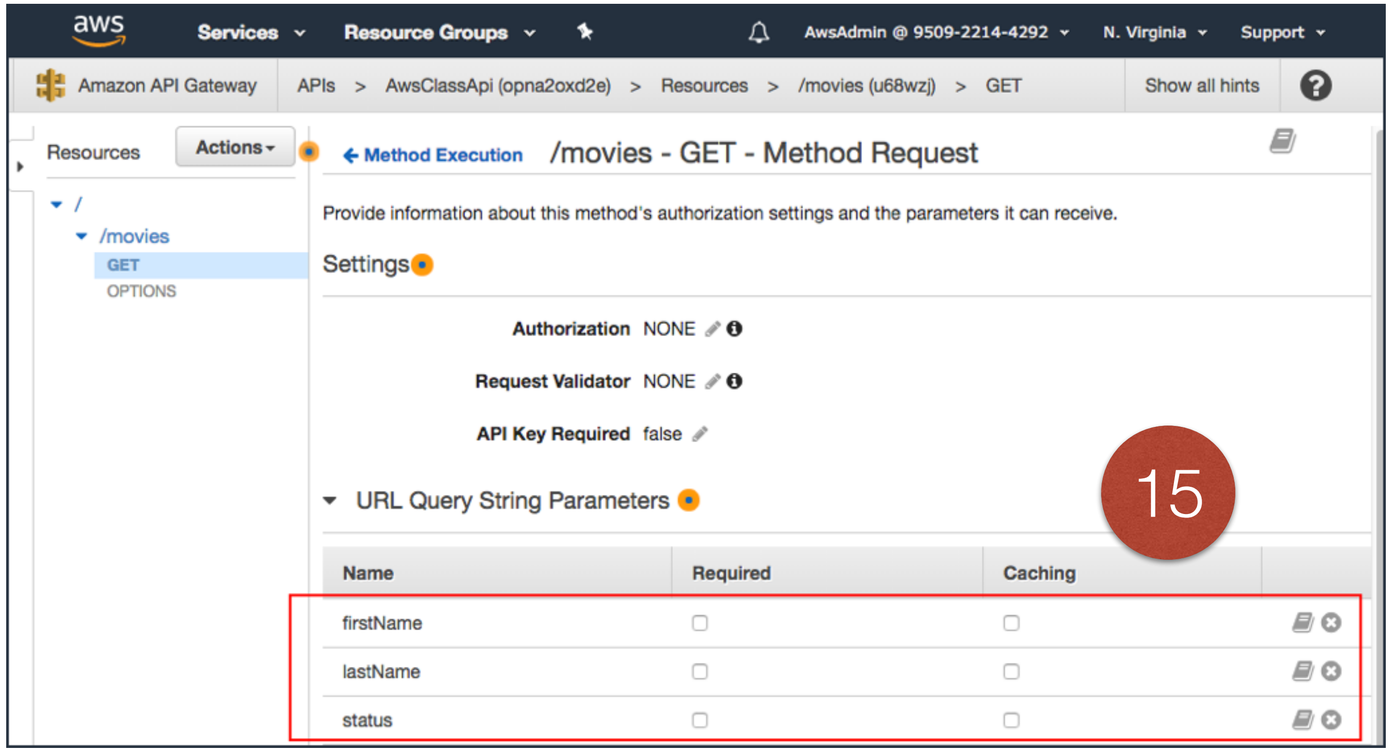


Creating API GET Endpoint

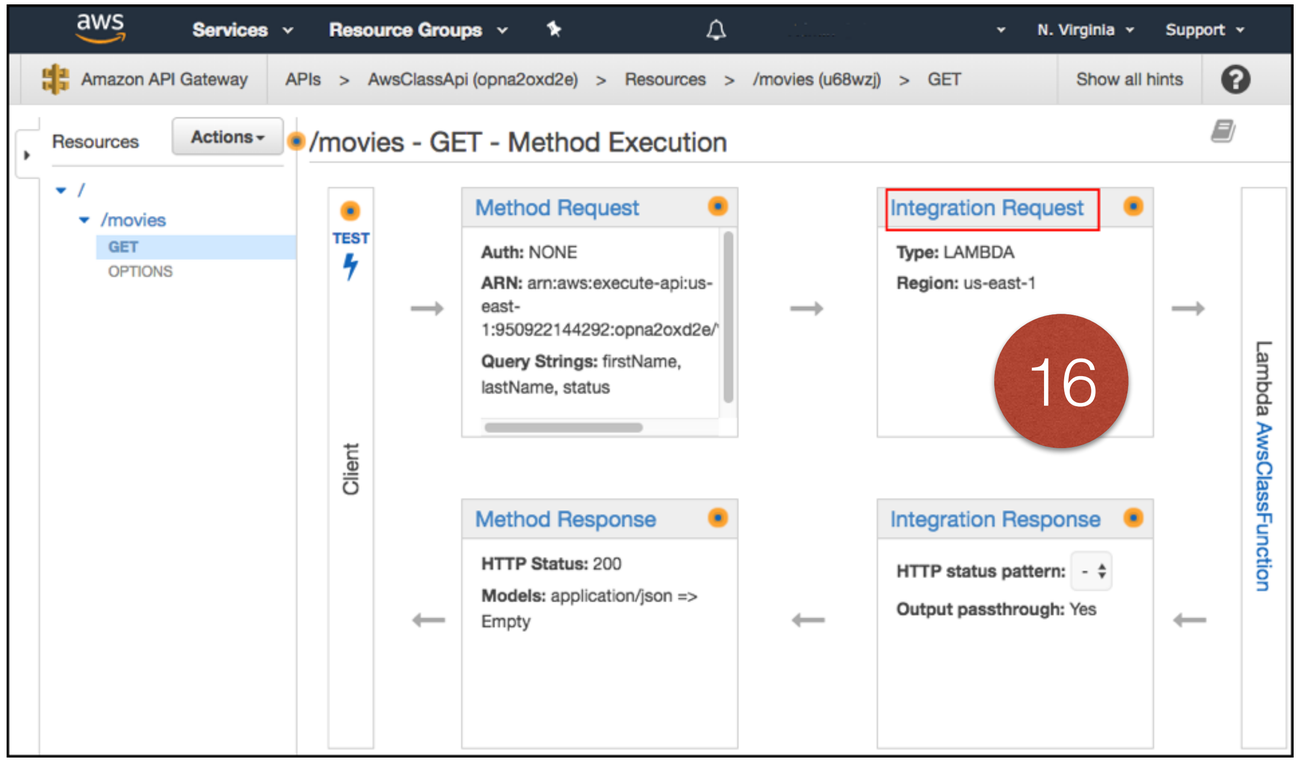


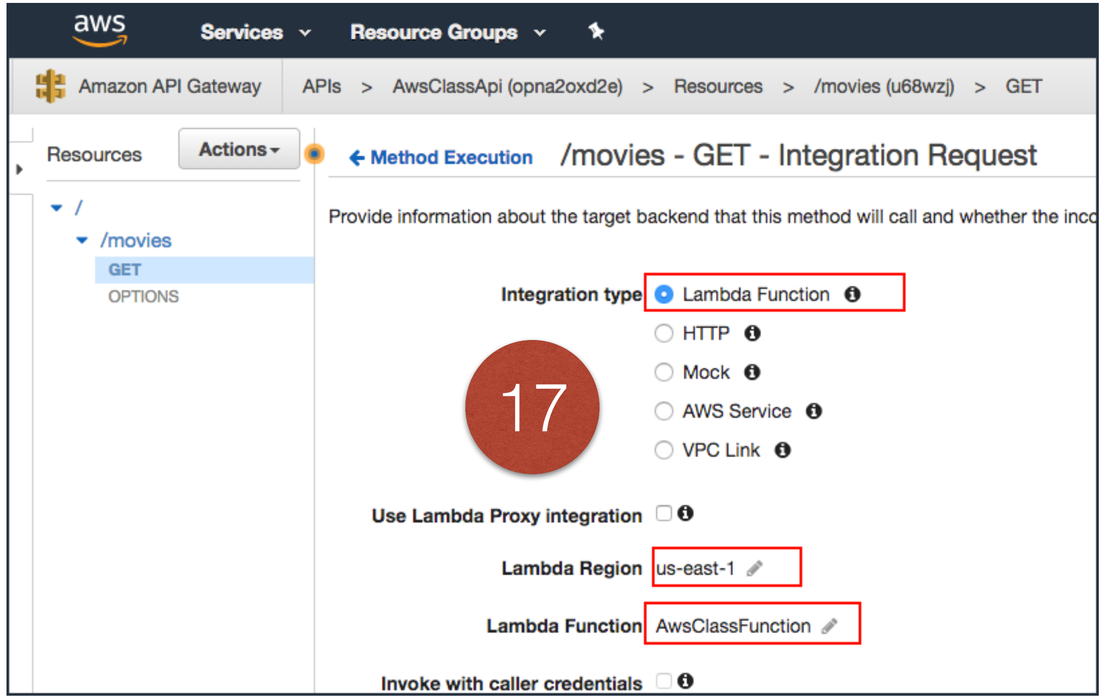


Creating API GET Endpoint

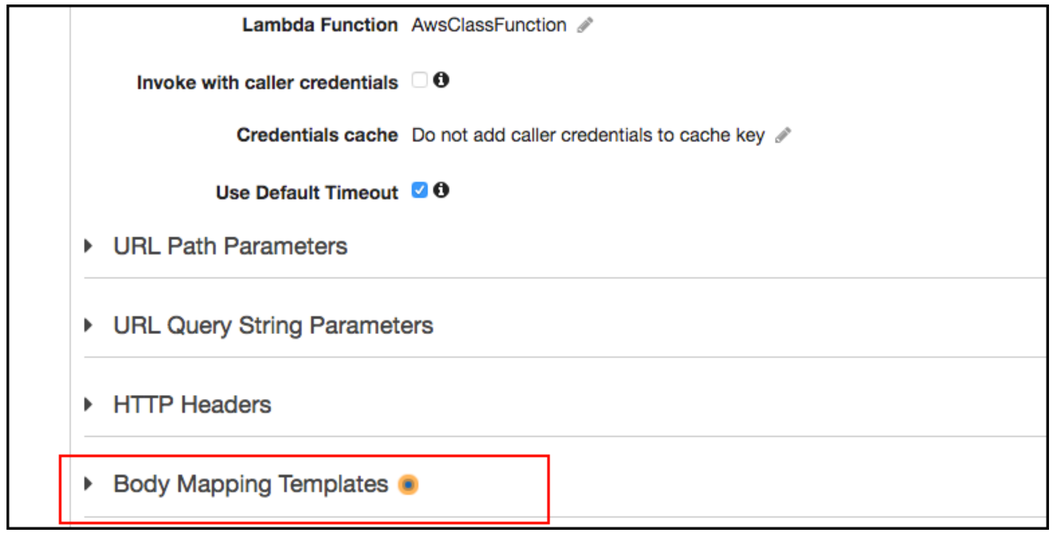


Creating API GET Endpoint

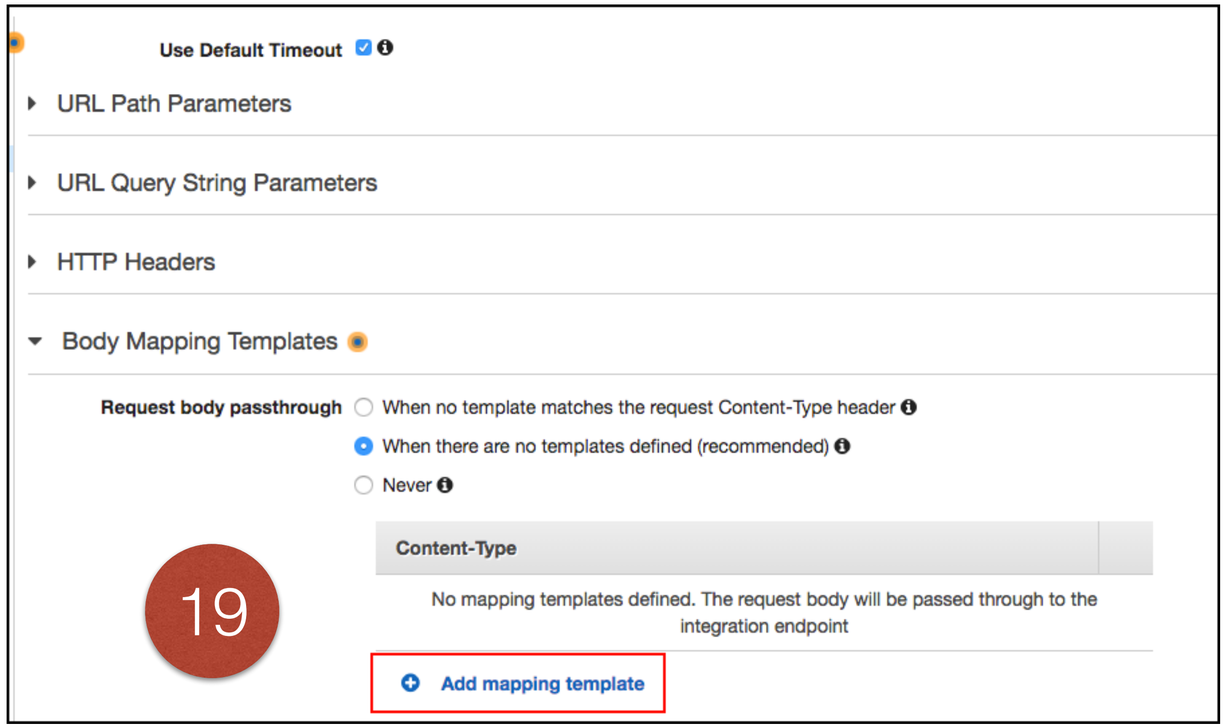




Creating API GET Endpoint

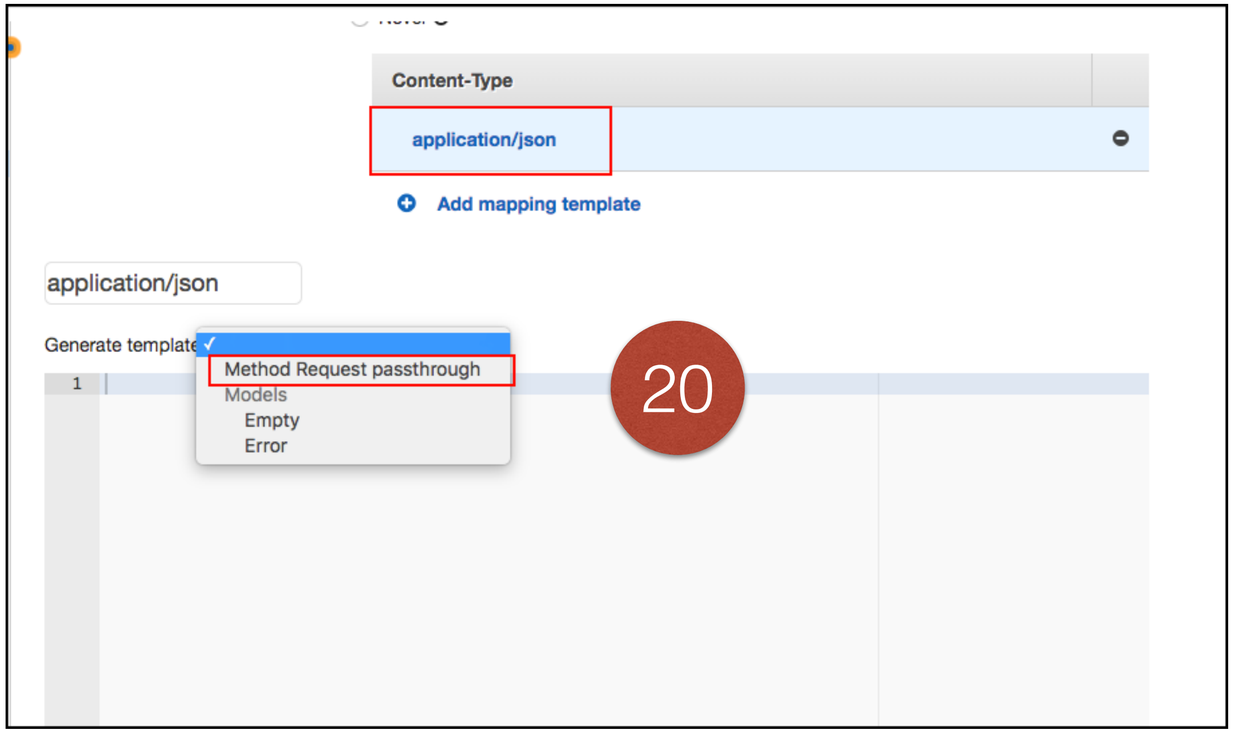


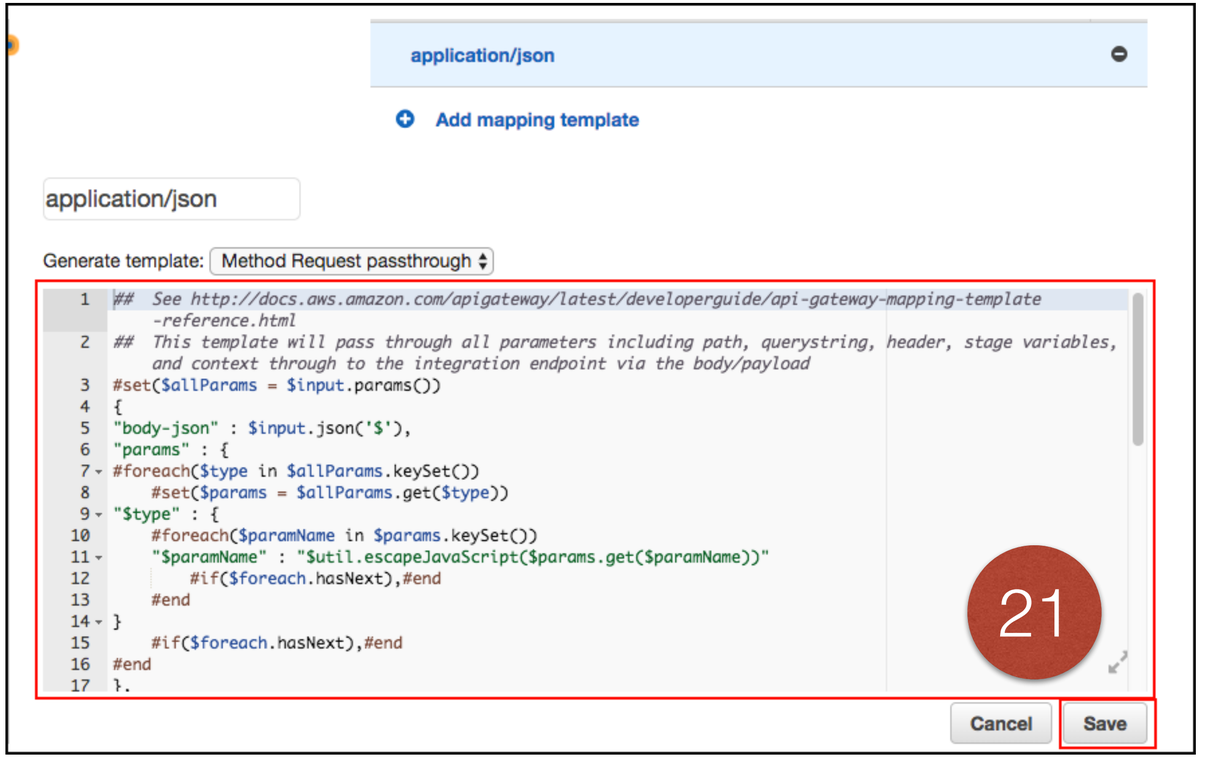
18



19

Creating API GET Endpoint

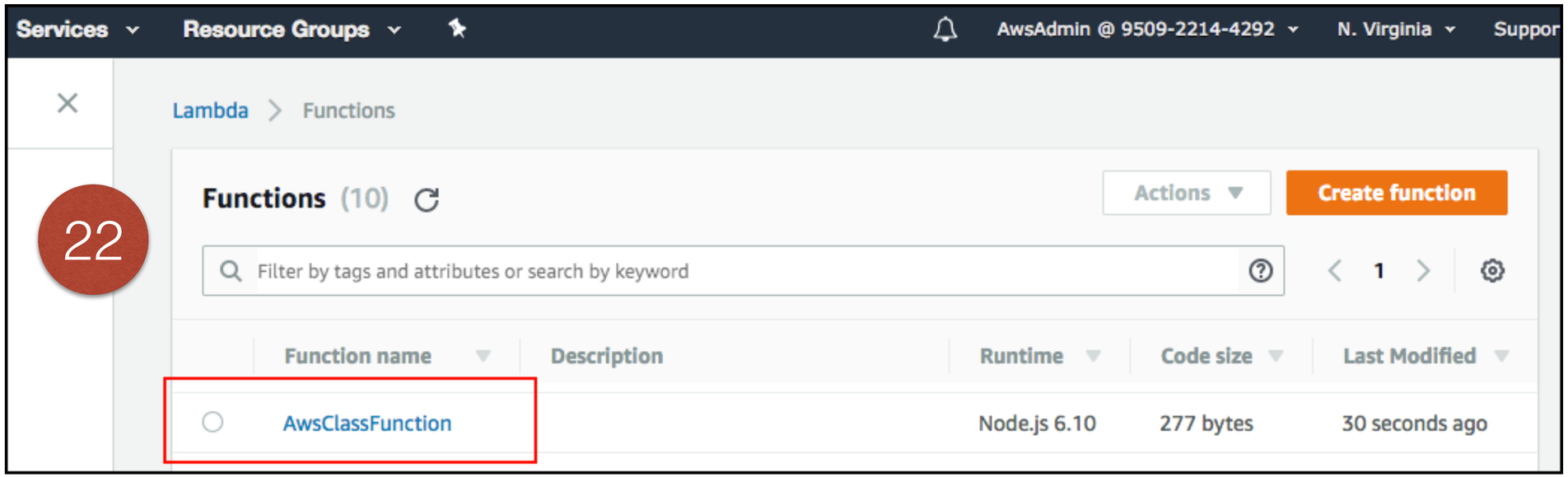




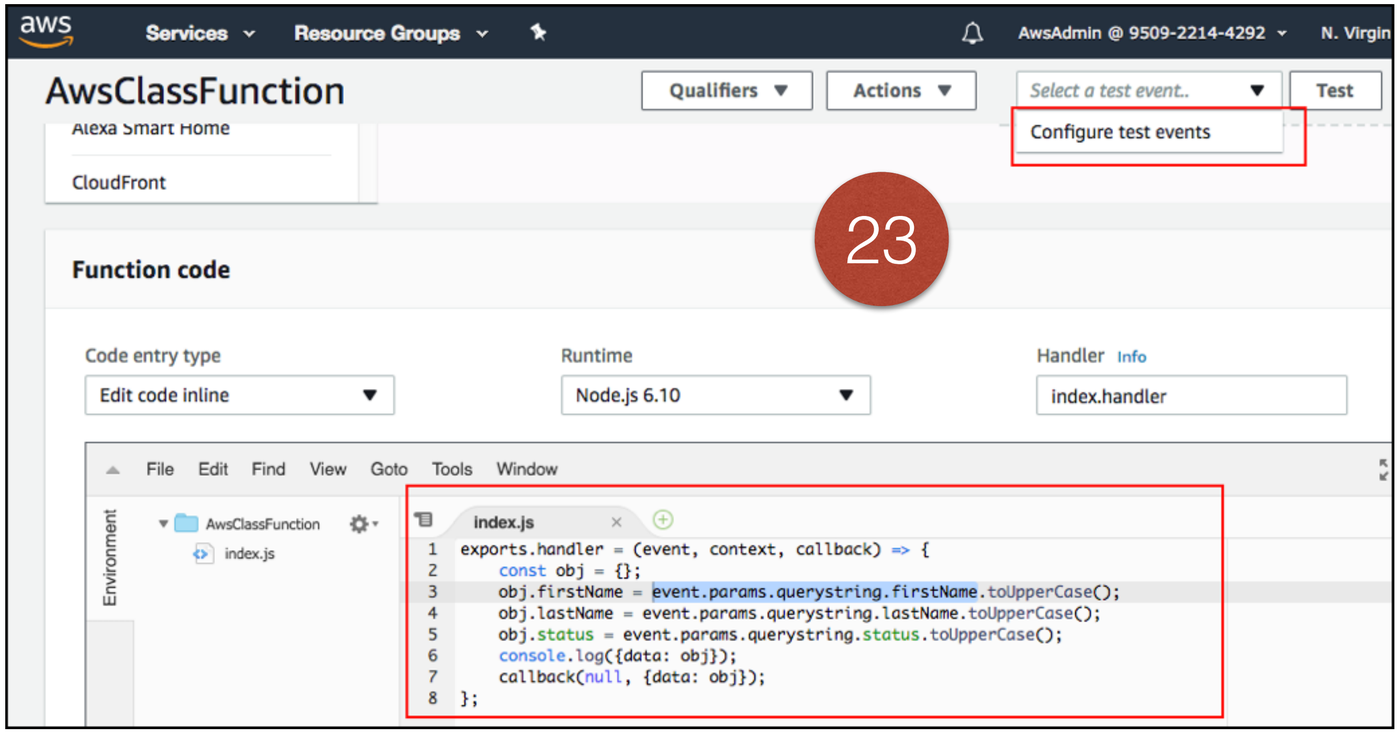
Binding API Gateway to Lambda

Mapping Template Info

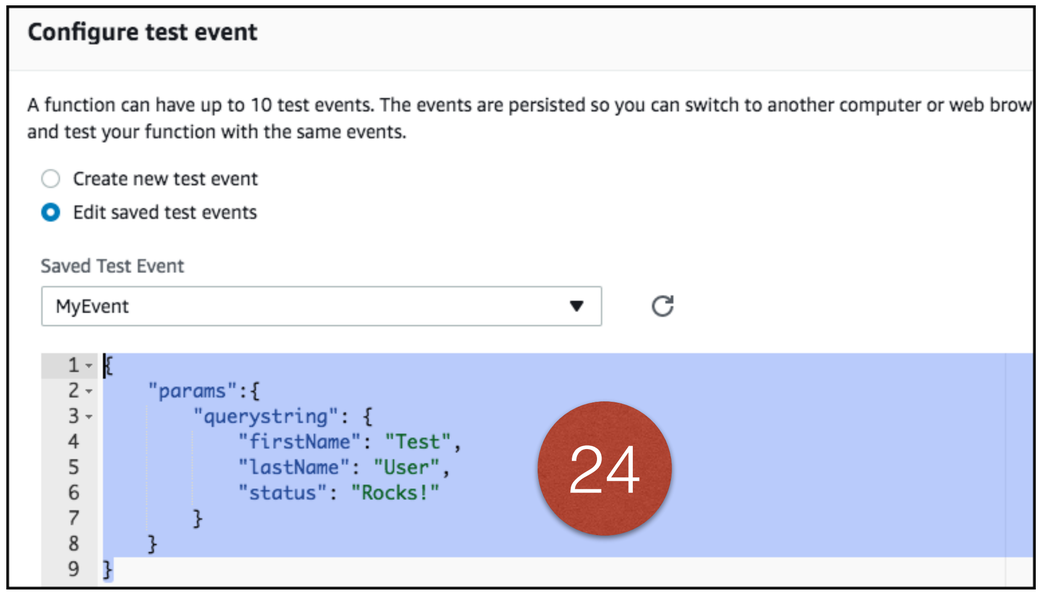
http://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-mapping-template-reference.html



Refactor Lamba

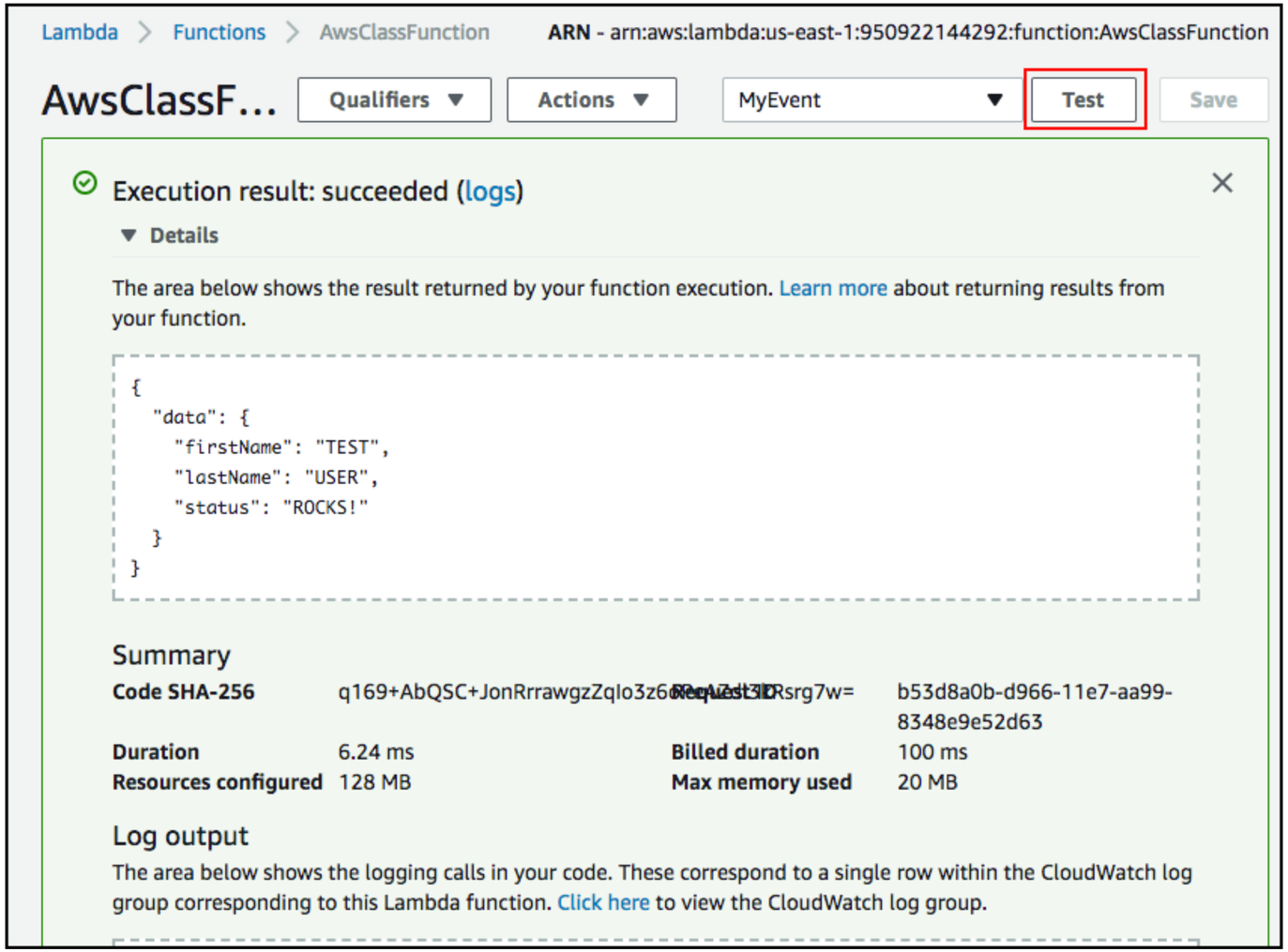


https://github.com/reselbob/CDAwsClass/blob/Session-3/AwsClassLambda.js



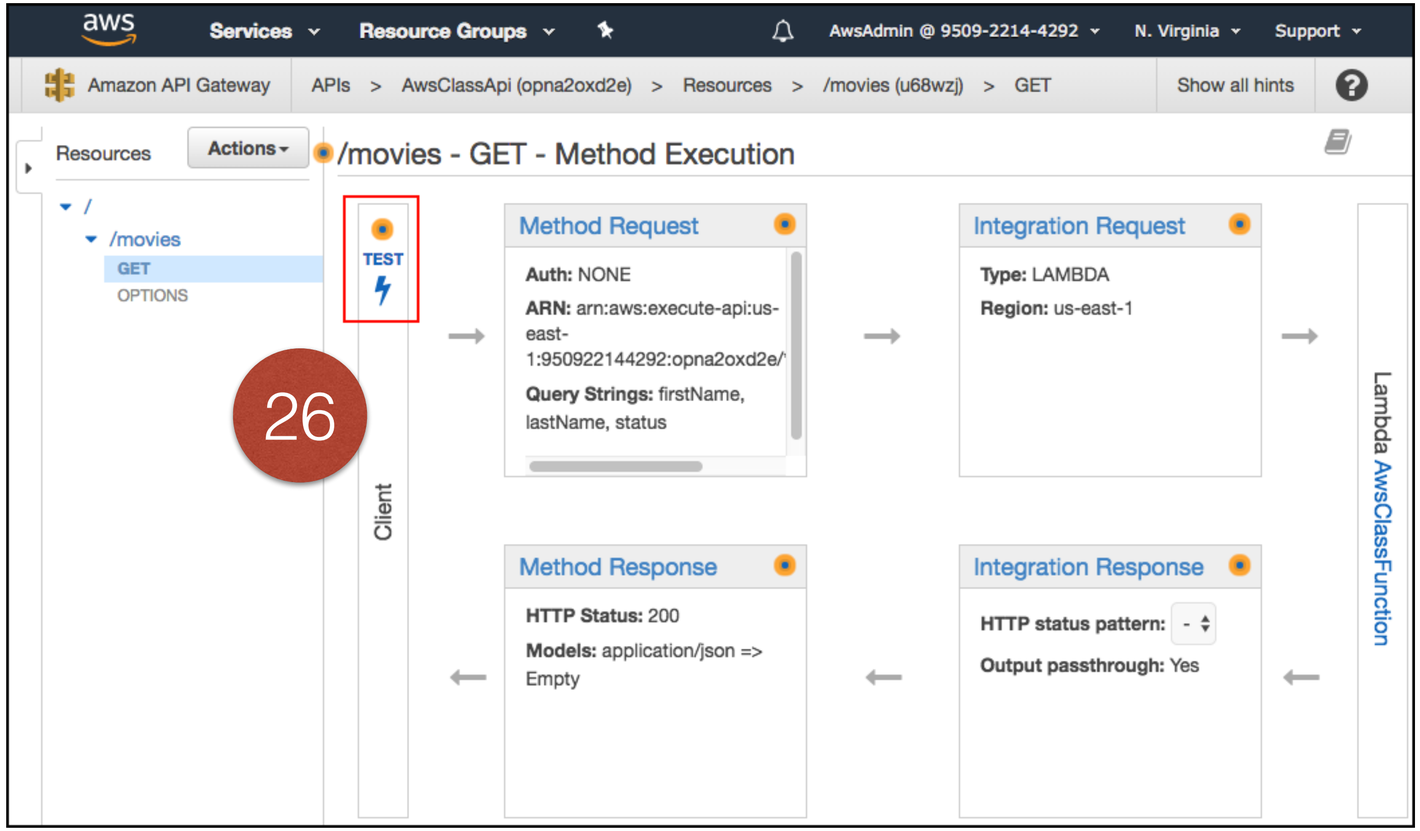
https://github.com/reselbob/CDAwsClass/blob/Session-3/AwsClassLambdaEvent.json

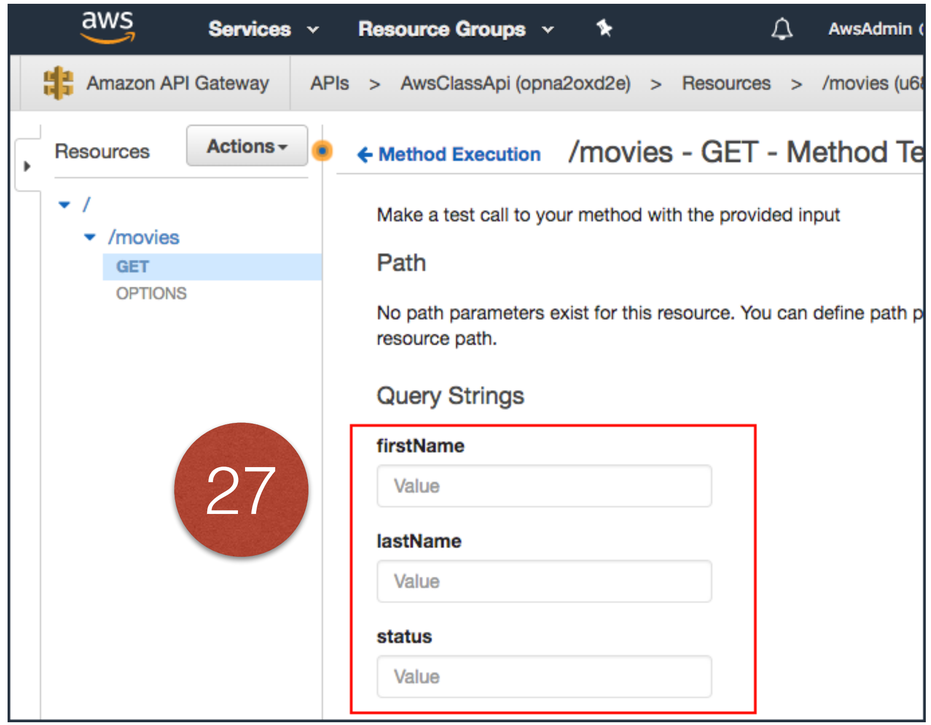
Refactor Lamba



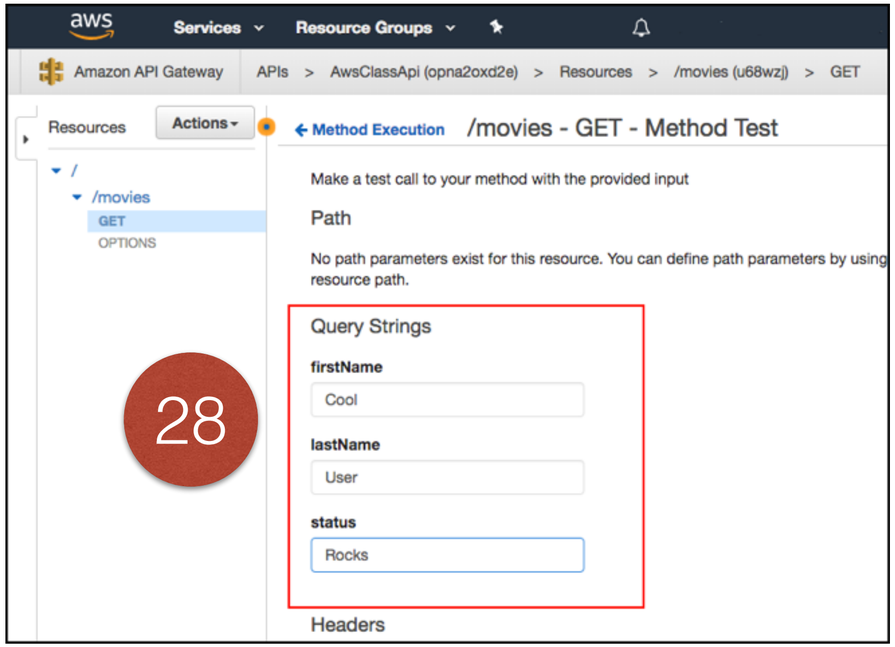
25

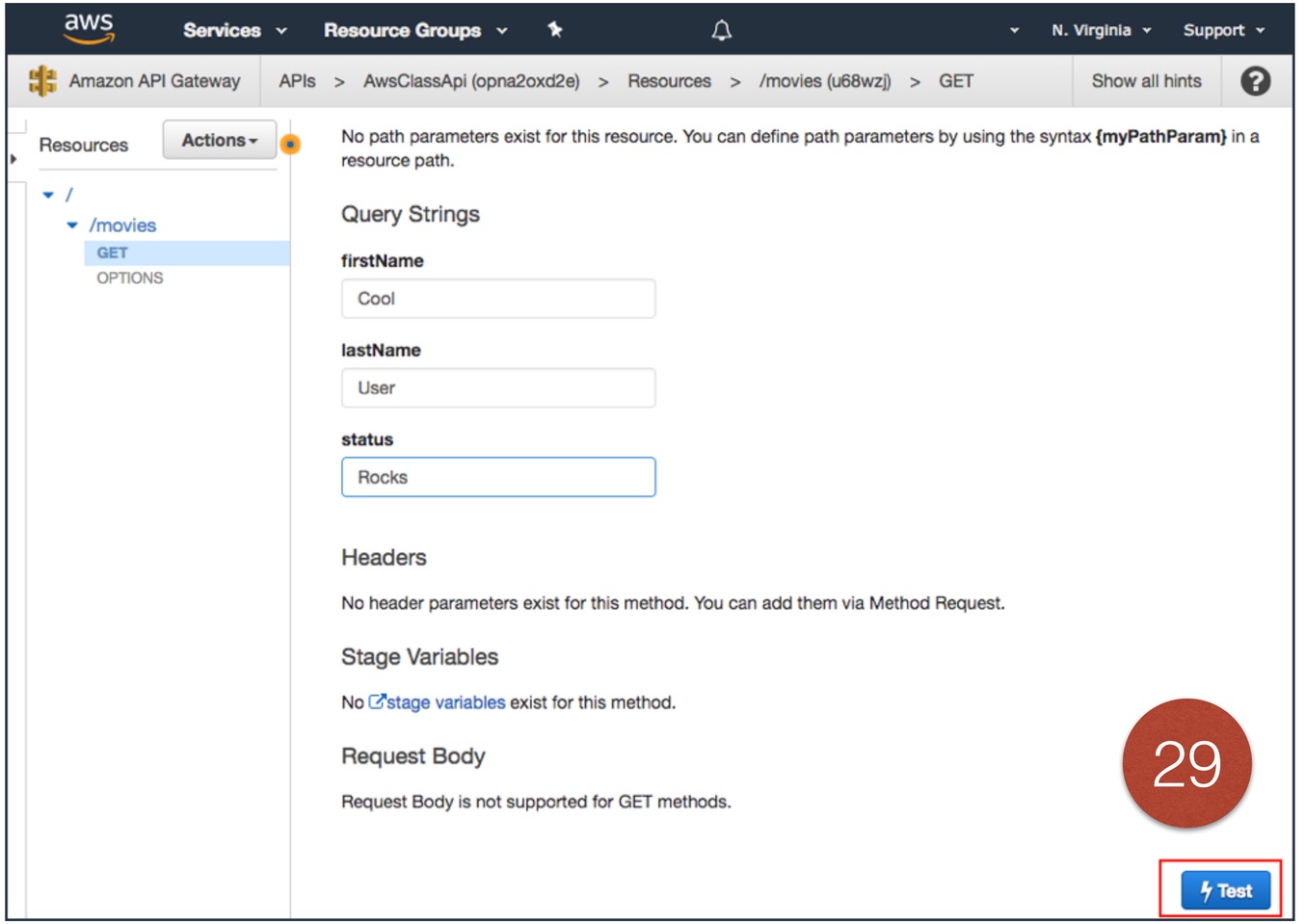
Testing the API





Testing the API





Testing the API

30

Request: /movies?lastName=Cool&status=User&firstName=Rocks

Status: 200

Latency: 381 ms

Response Body


```
{
  "data": {
    "firstName": "ROCKS",
    "lastName": "COOL",
    "status": "USER"
  }
}
```

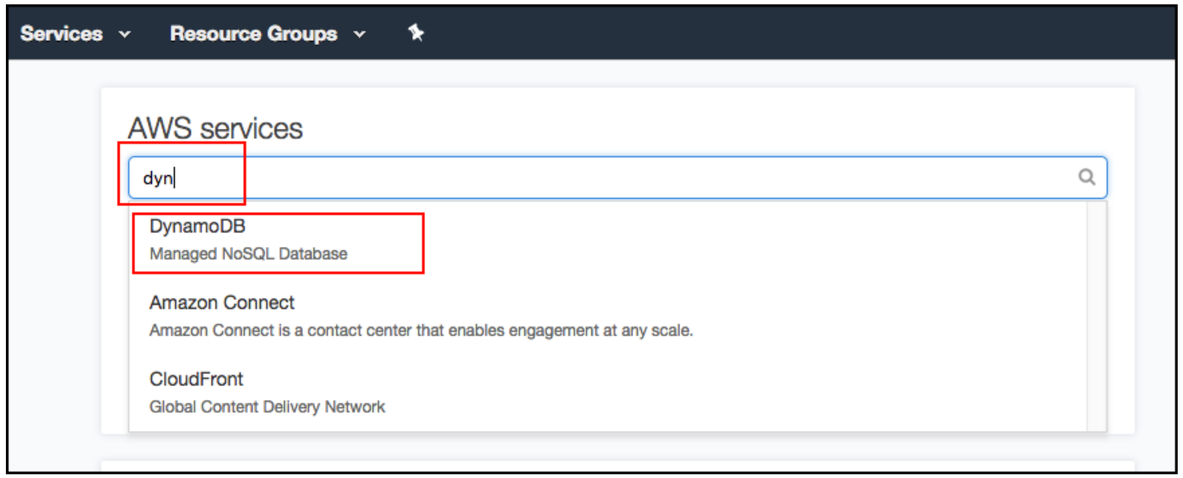

Response Headers


```
{"X-Amzn-Trace-Id":"sampled=0;root=1-5a2618d7-efde4c7547217b6929a63bee","Content-Type":"appli
cation/json"}
```

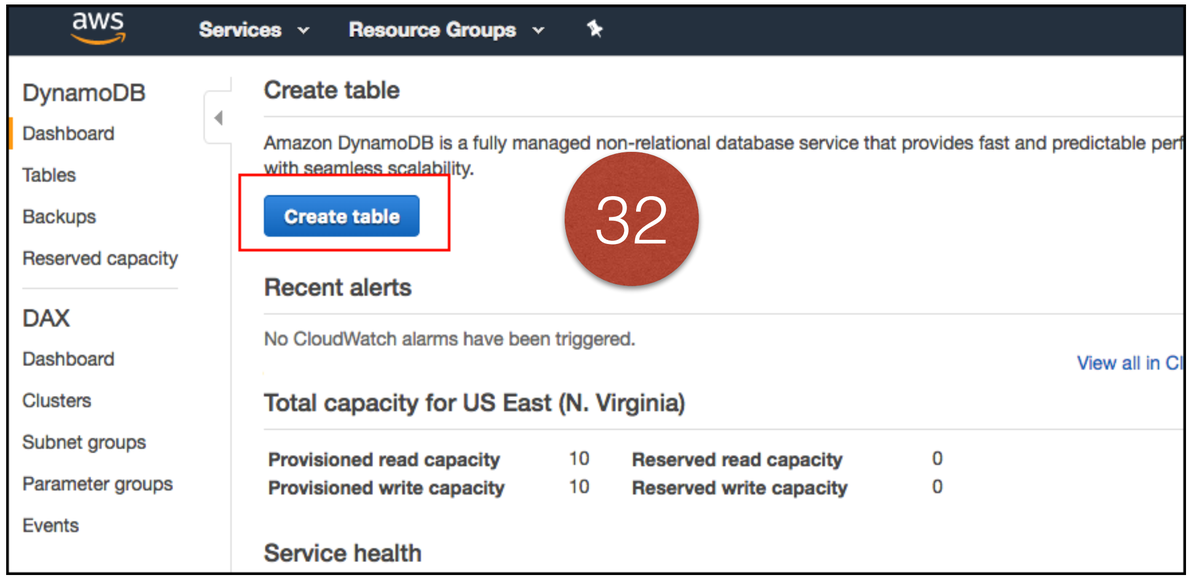

Logs

```
Execution log for request test-request
```

Creating the DynamoDB Table



31



Creating the DynamoDB Table

Create DynamoDB table

Tutorial

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

33

Table name* Ratings

Primary key* Partition key

movieTitle String

Add sort key

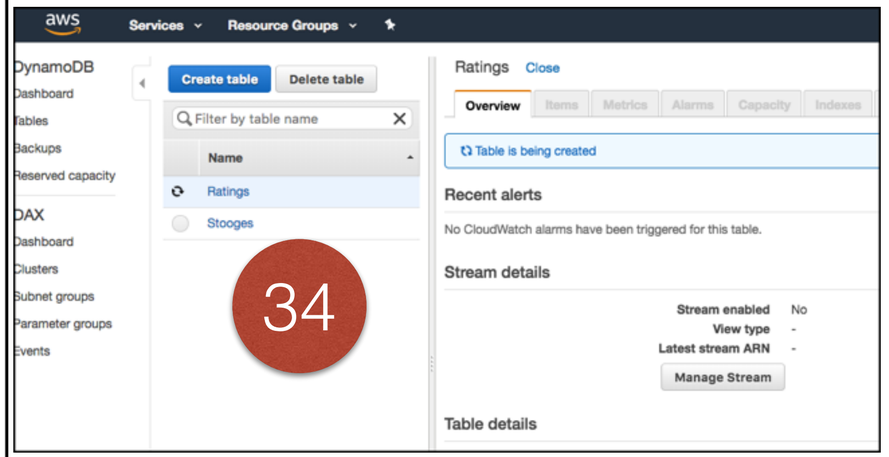
Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- On-Demand Backup and Restore Enabled NEW!

Cancel **Create**



Questions and Answers

End of Day 1
Review and Evaluation

Please take the Day 1 Evaluation:

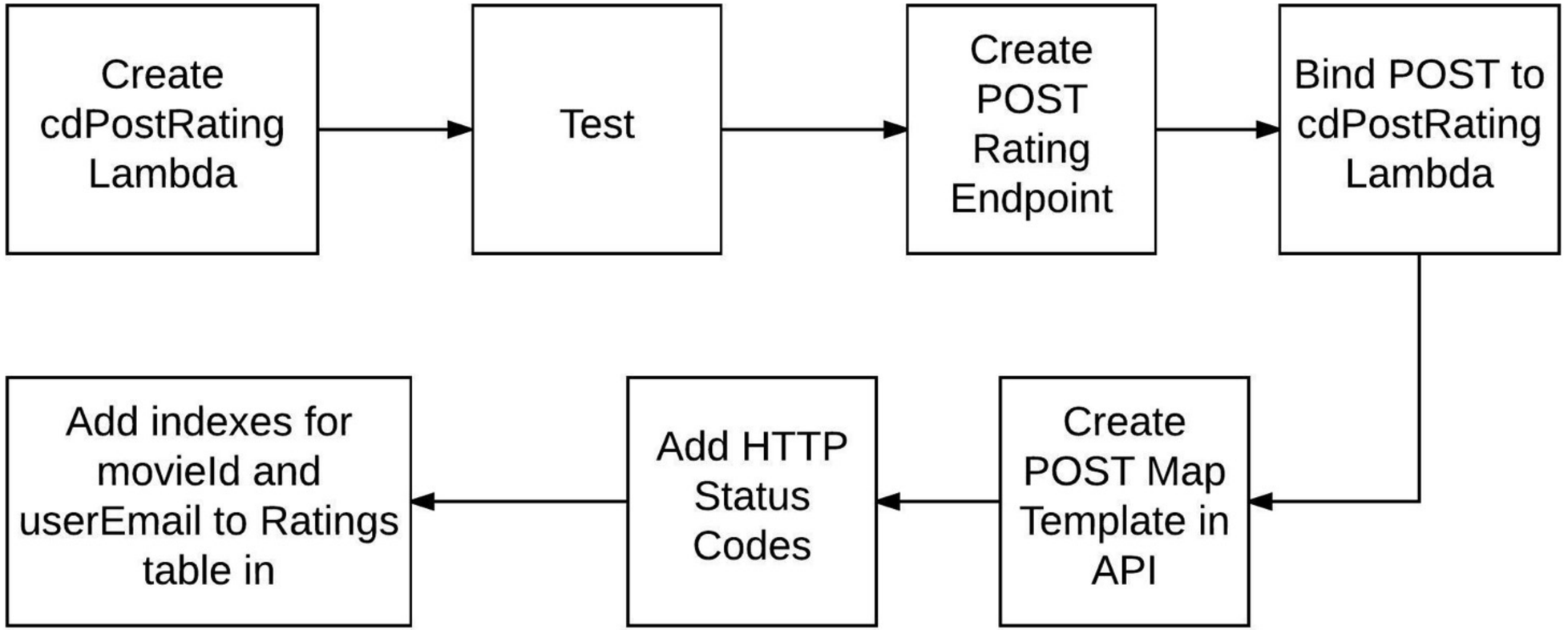
https://www.surveymonkey.com/r/25H6BSB

Bob Reselman
reselbob@gmail.com

Agenda for Day 2

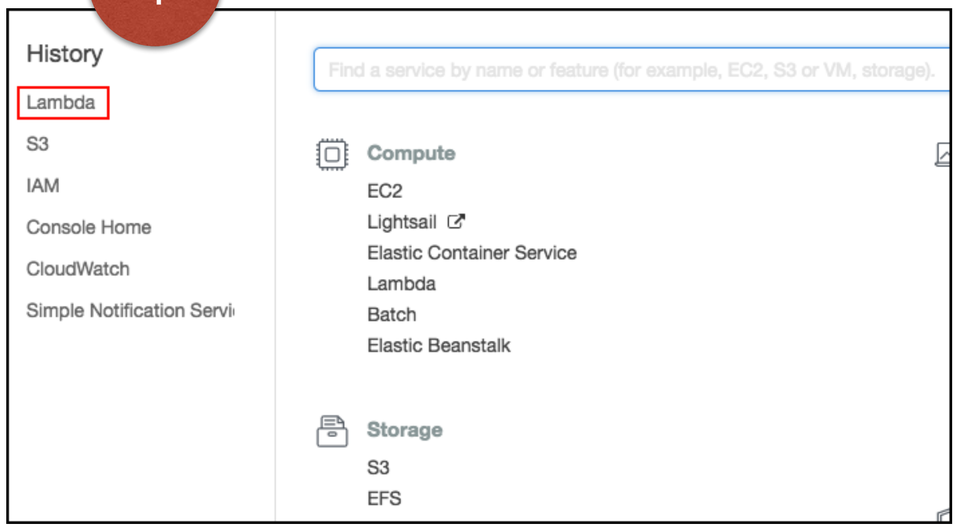
- **Session 4**
  - Creating the POST Rating API Endpoint
  - Creating the postRating Lambda Function
  - Creating the Rating Index in DynamoDB
  - Supporting Status Codes
- **Session 5**
  - Creating the getMovies Lambda Function
  - Refactor the GET Movies API Endpoint
  - Creating the getRating Lambda Function
  - Creating the GET Ratings API Endpoint
  - Publishing the API to a Test Deployment
- **Session 6**
  - Refactoring the Movie Rater UI
  - Creating and Binding the SNS Topic
  - Subscribing to the MovieRater Topic

Session 4

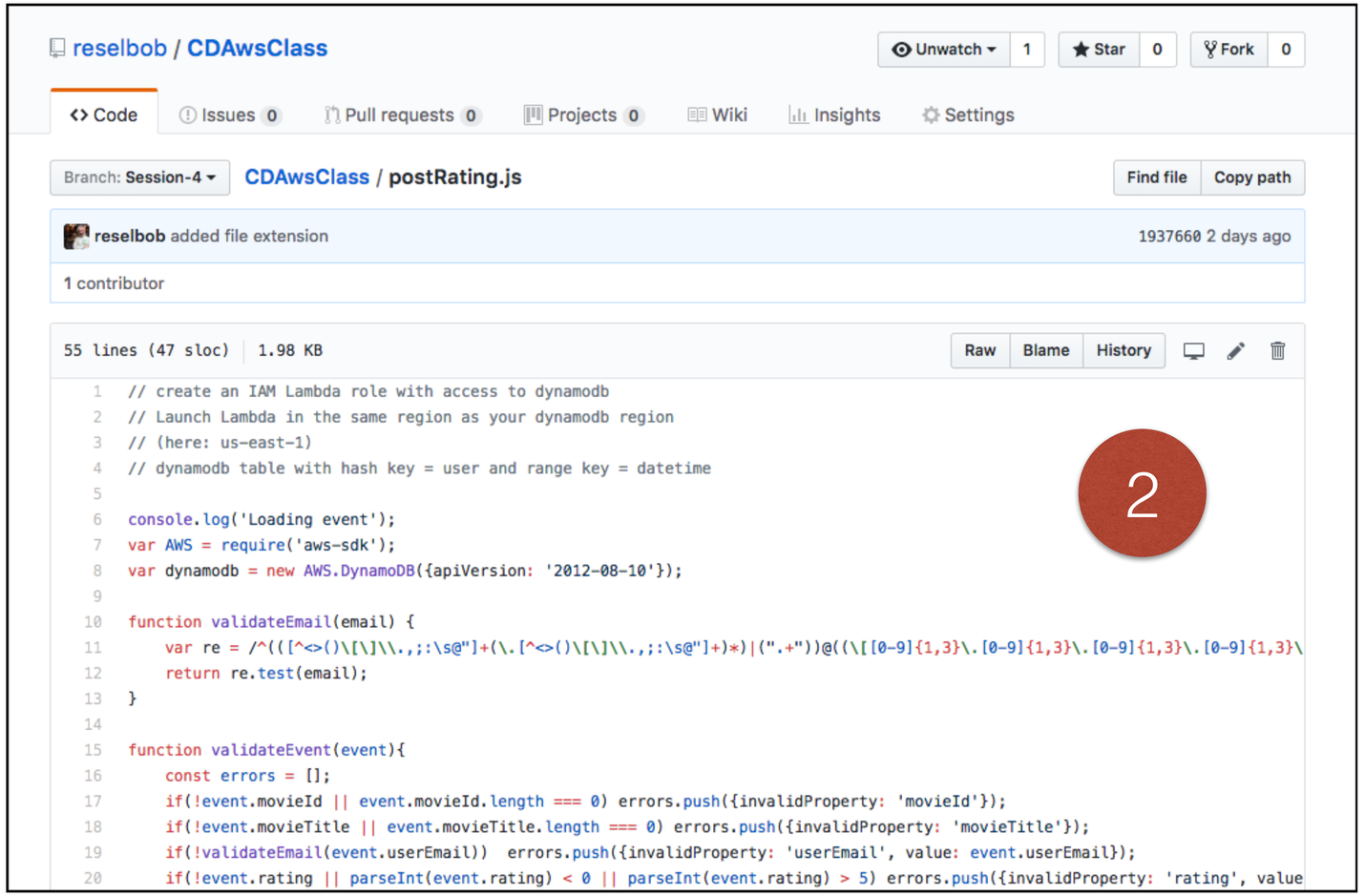

Create cdPostRating Lambda
Test
Create POST Rating Endpoint
Bind POST to cdPostRating Lambda
Create POST Map Template in API
Add HTTP Status Codes
Add indexes for movieId and userEmail to Ratings table in

Create POST Rating Lambda

1

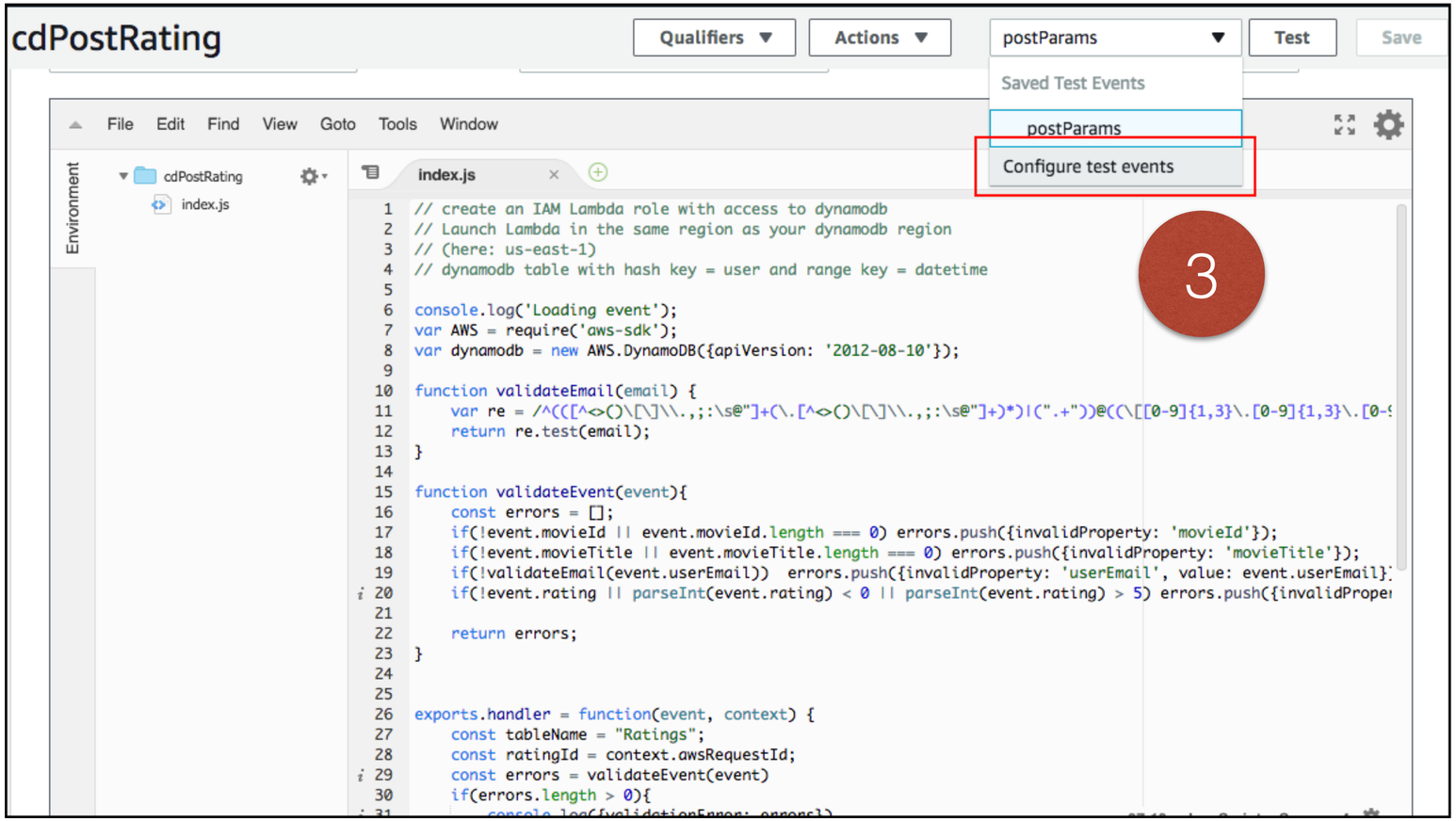


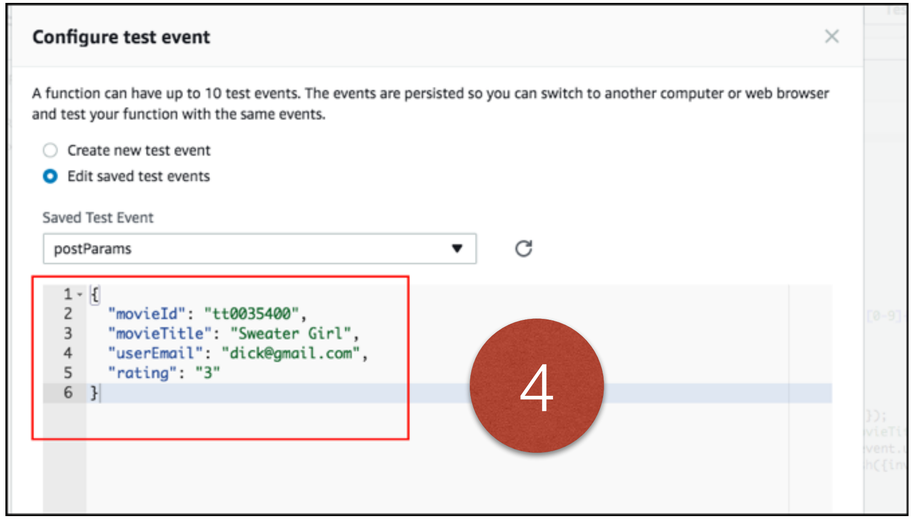
2



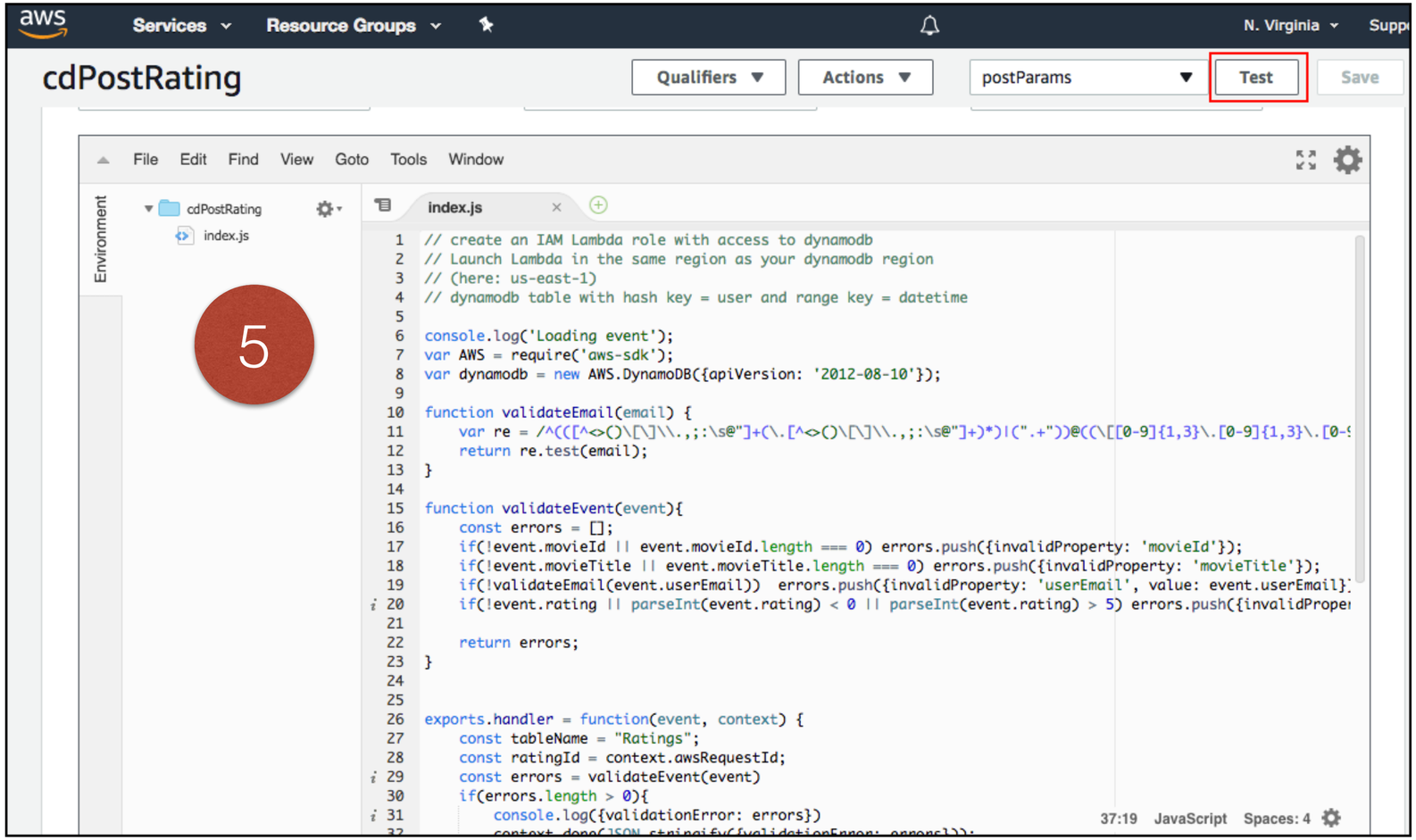
https://github.com/reselbob/CDAwsClass/blob/Session-4/postRating.js

Create POST Rating Lambda

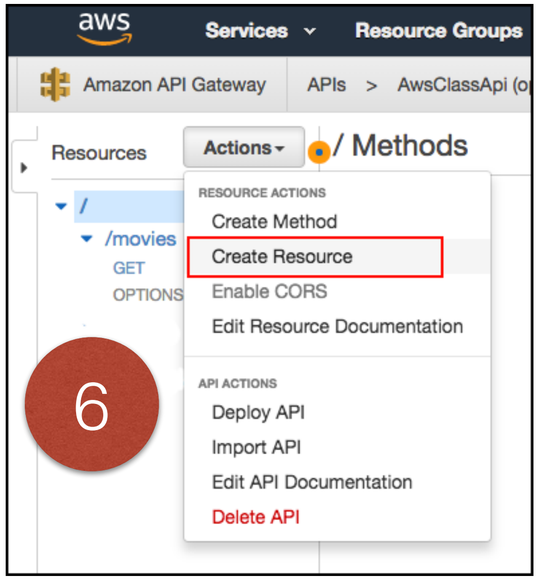


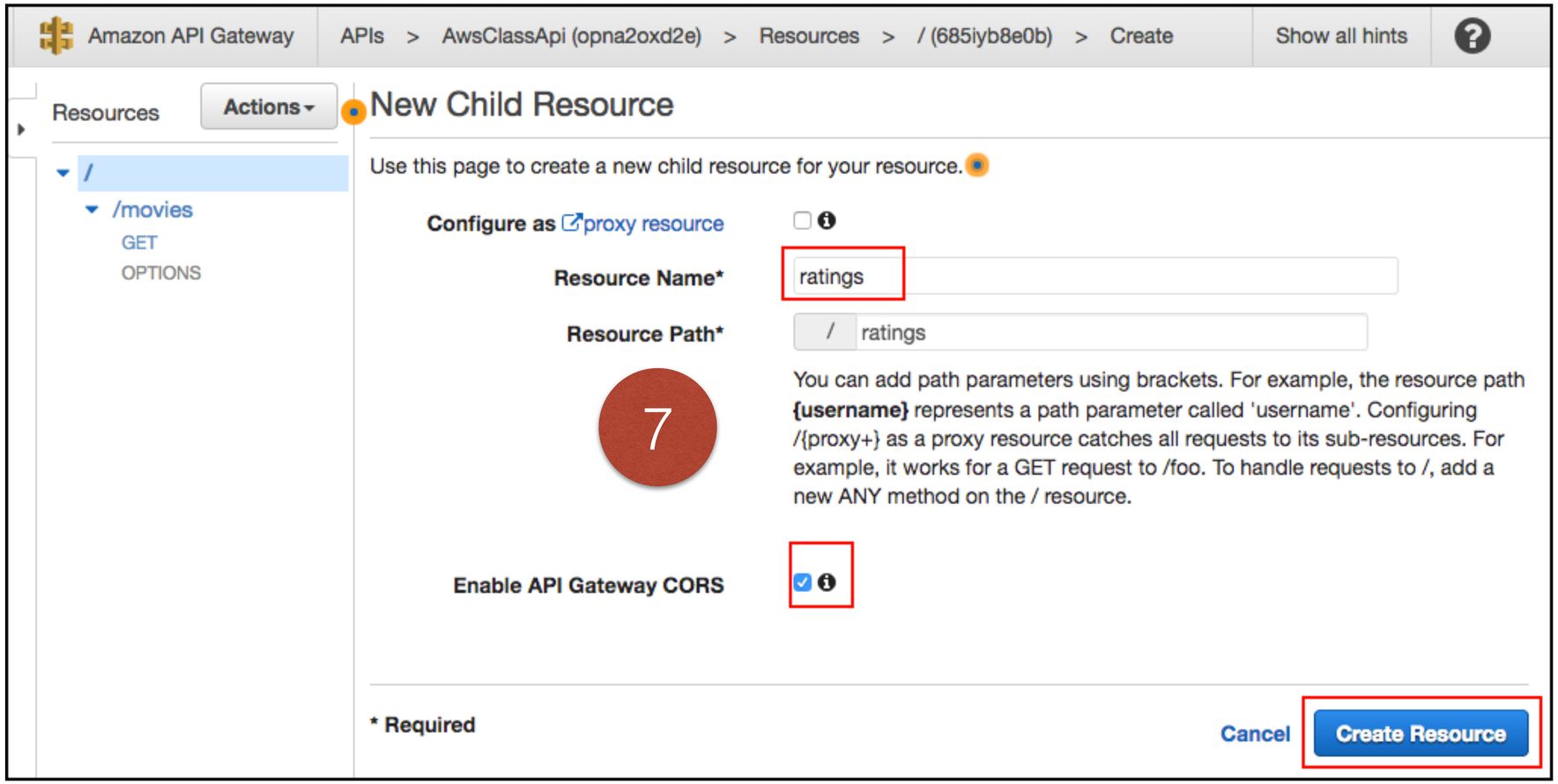


Test POST Rating Lambda

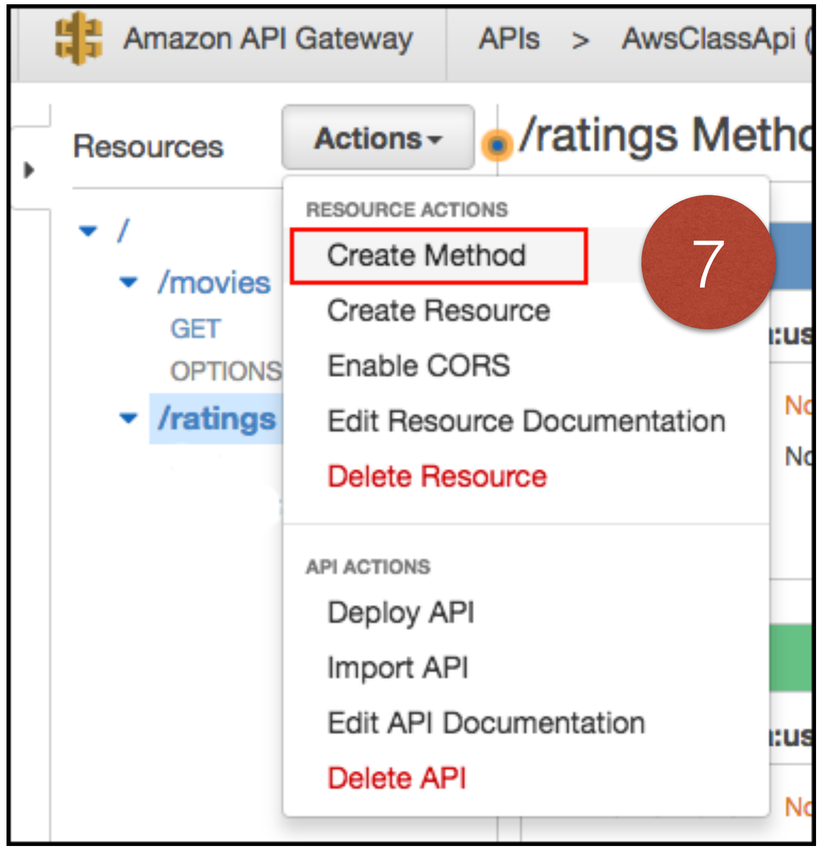


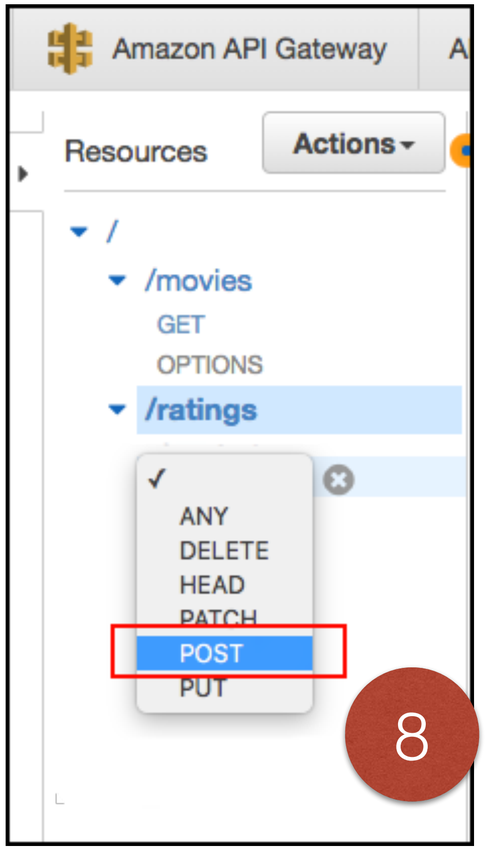
Create POST Rating Endpoint

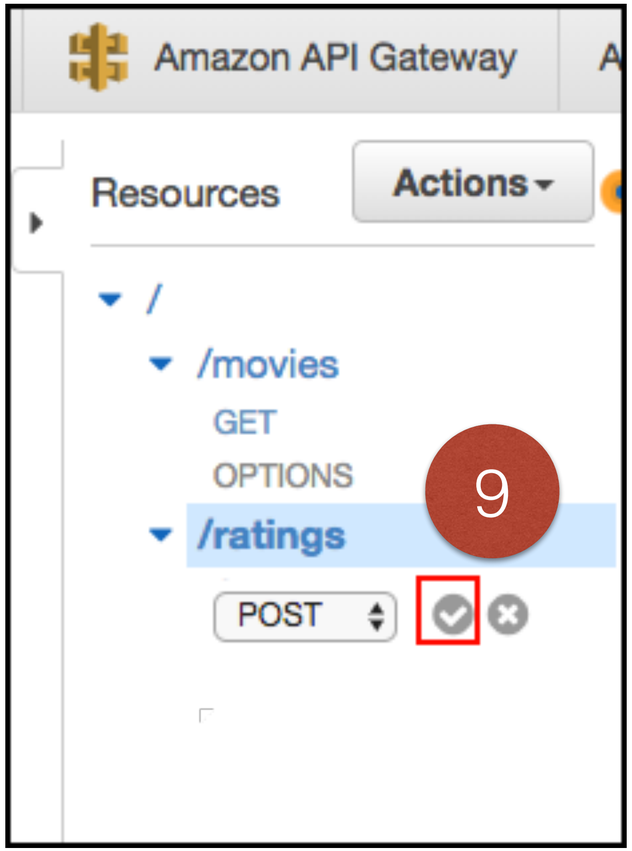




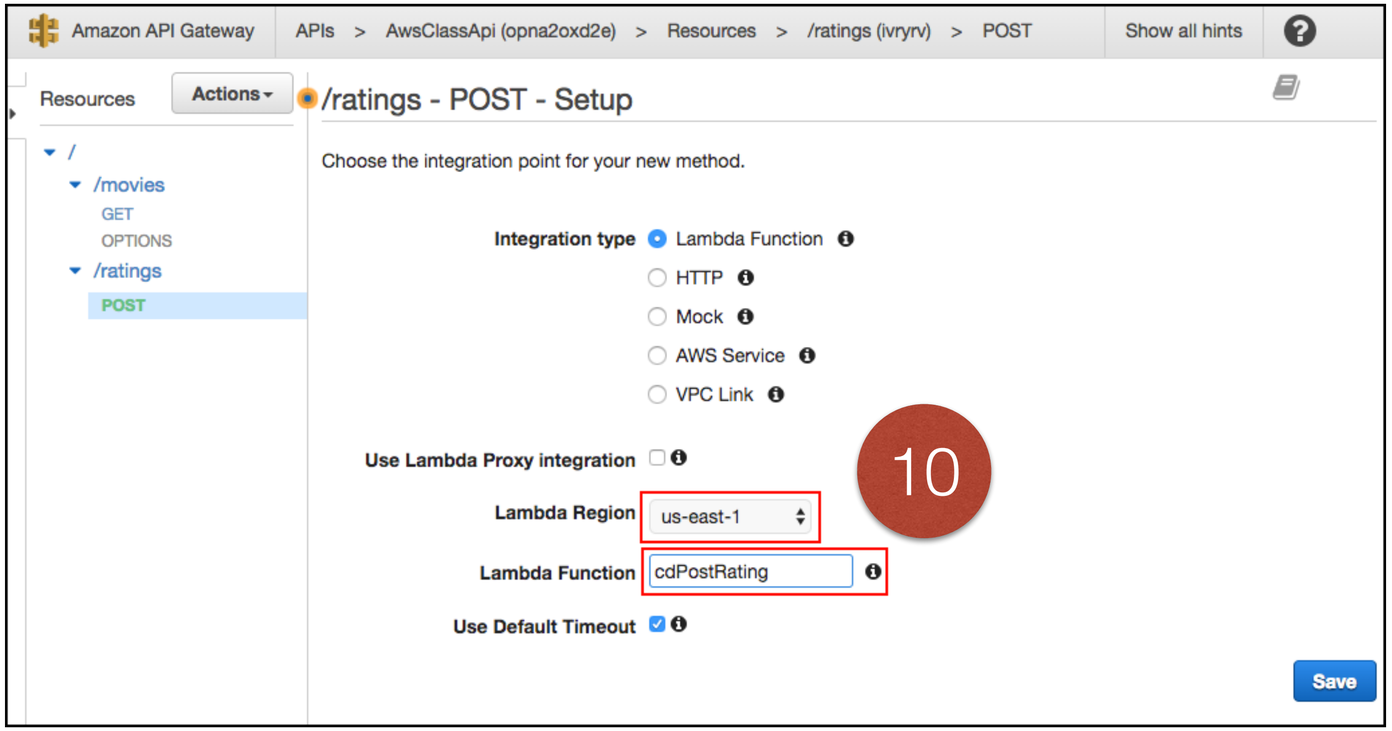
Create POST Rating Endpoint

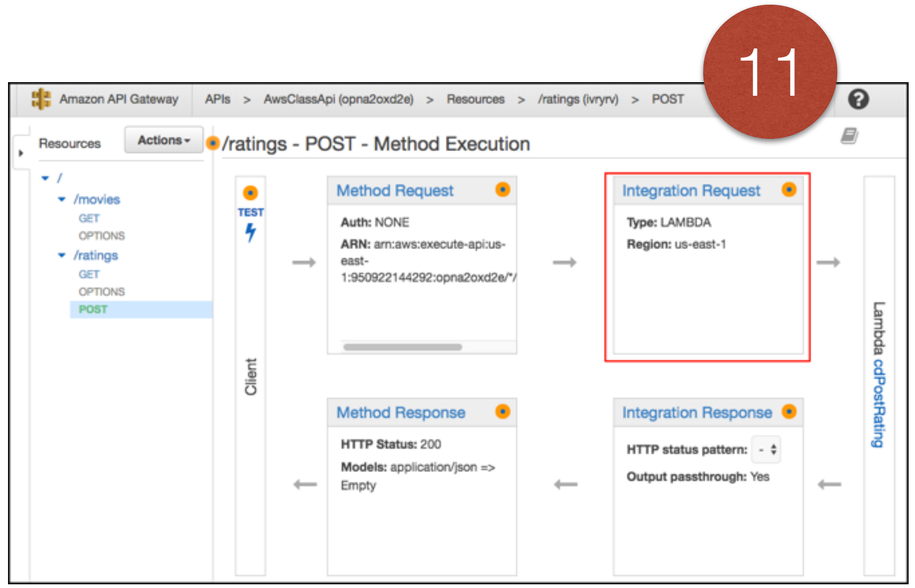




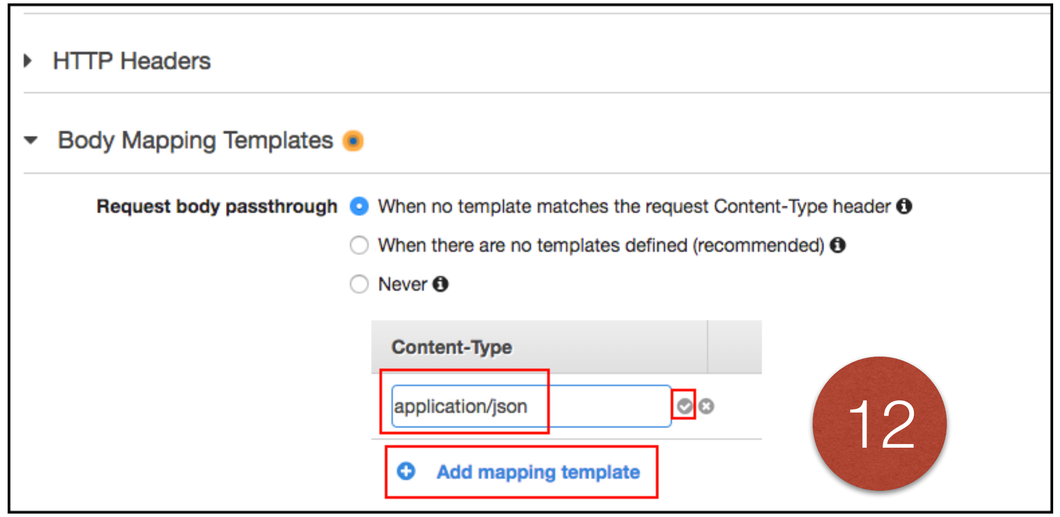


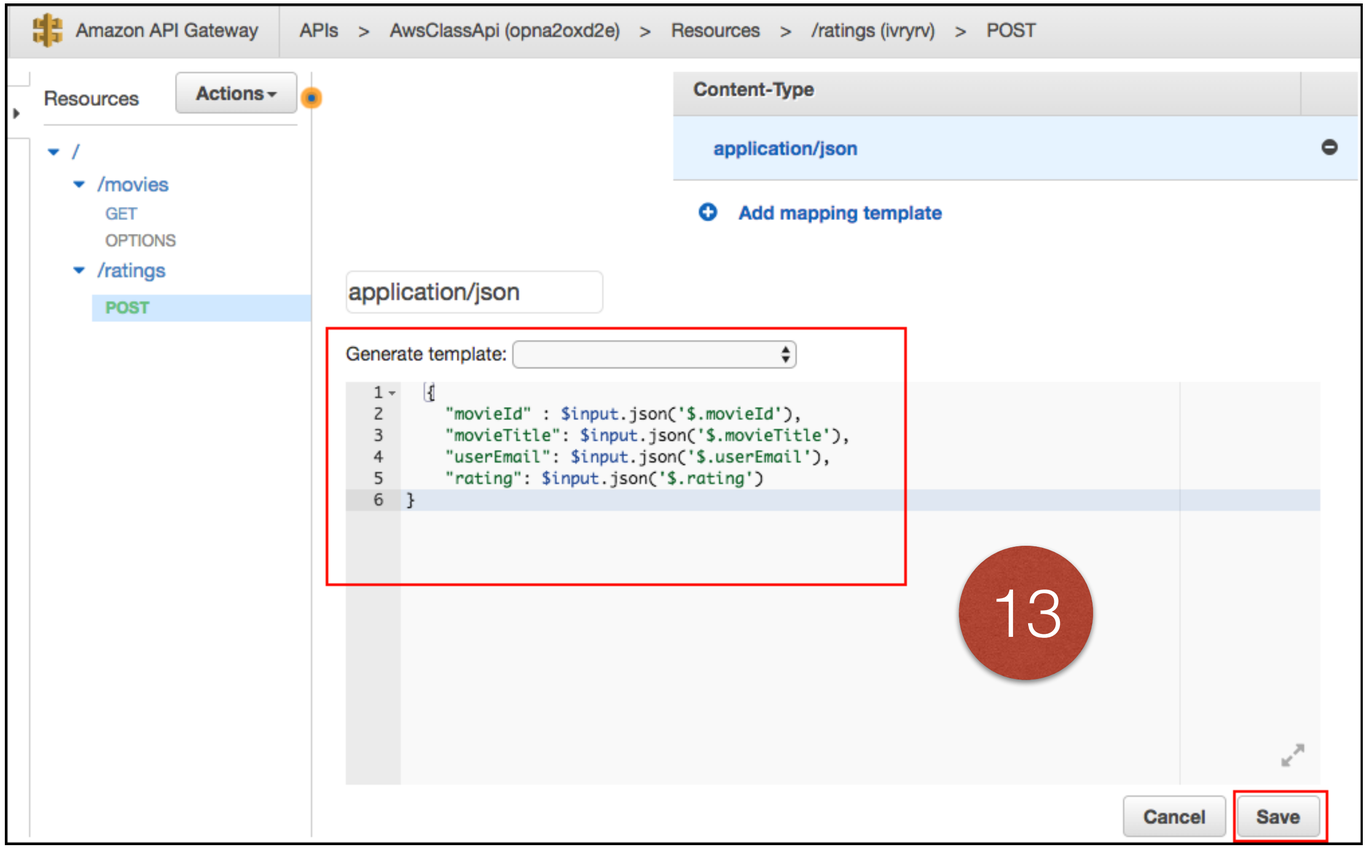
Create POST Rating Endpoint



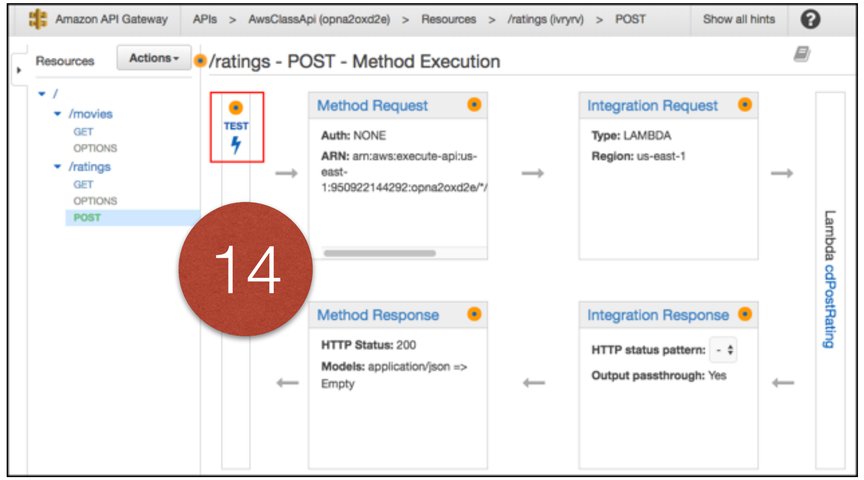


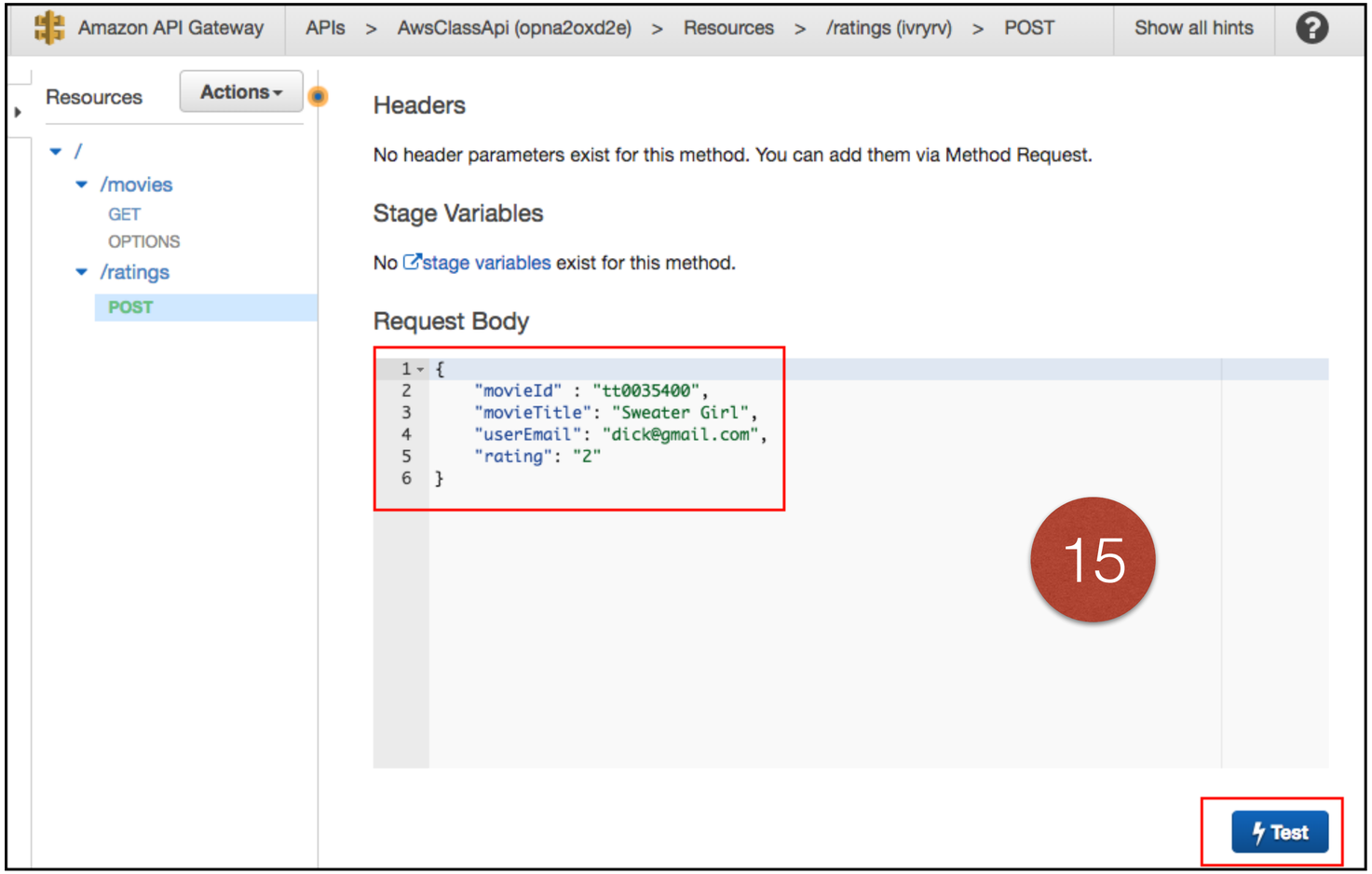
Create POST Rating Endpoint



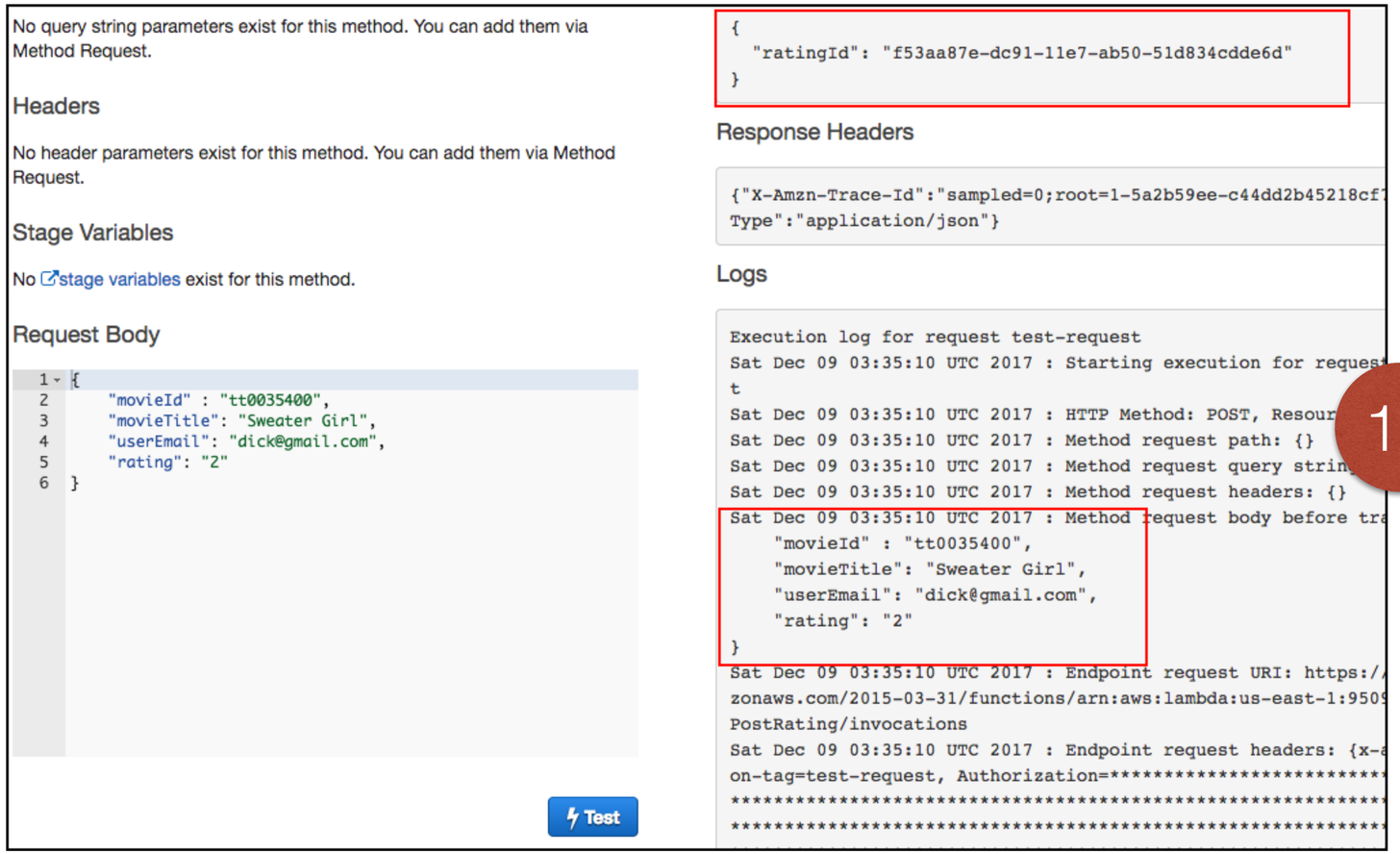


Create POST Rating Endpoint

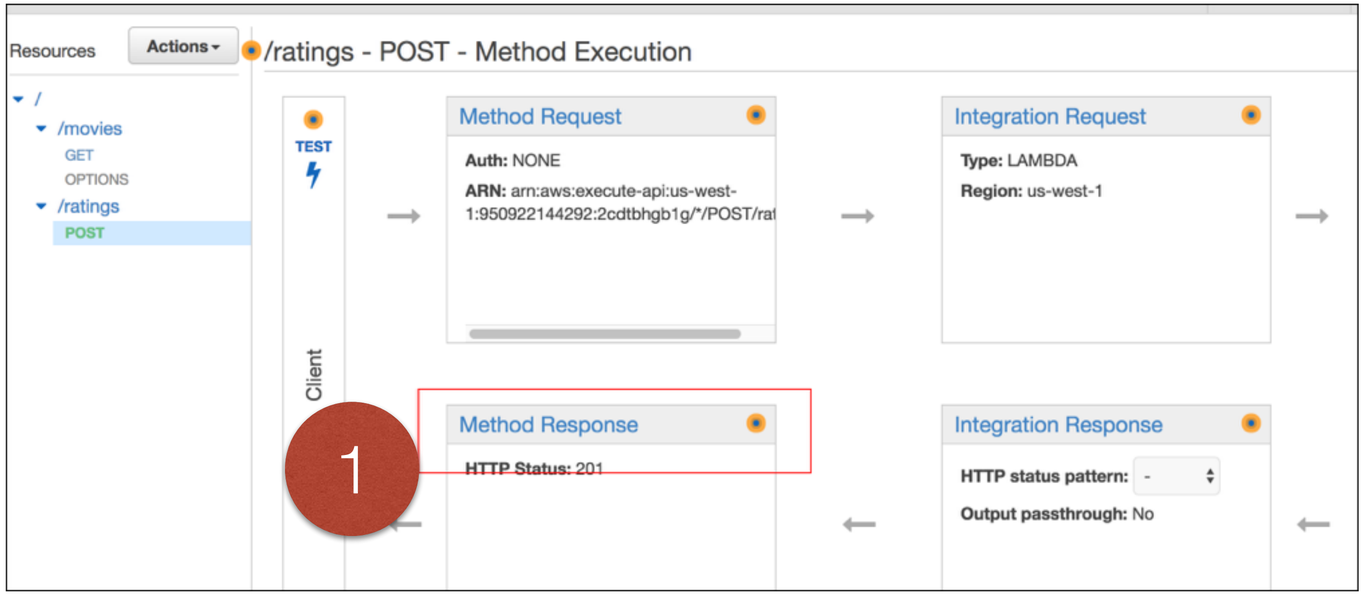


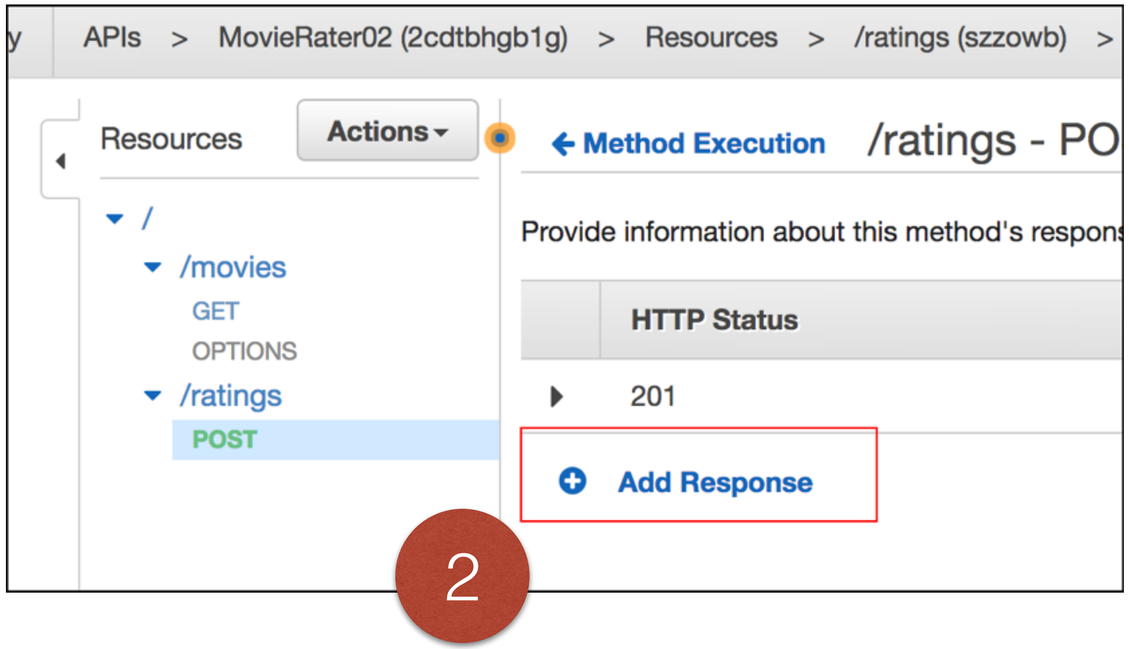


Create POST Rating Endpoint

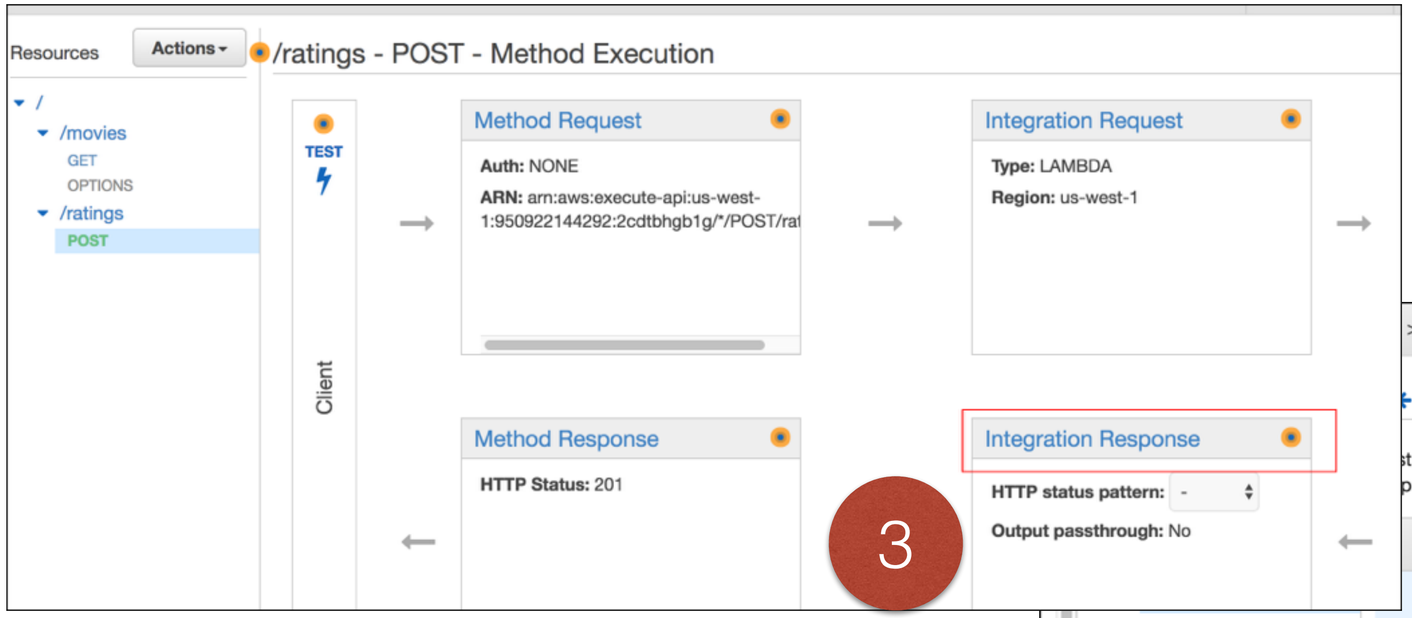


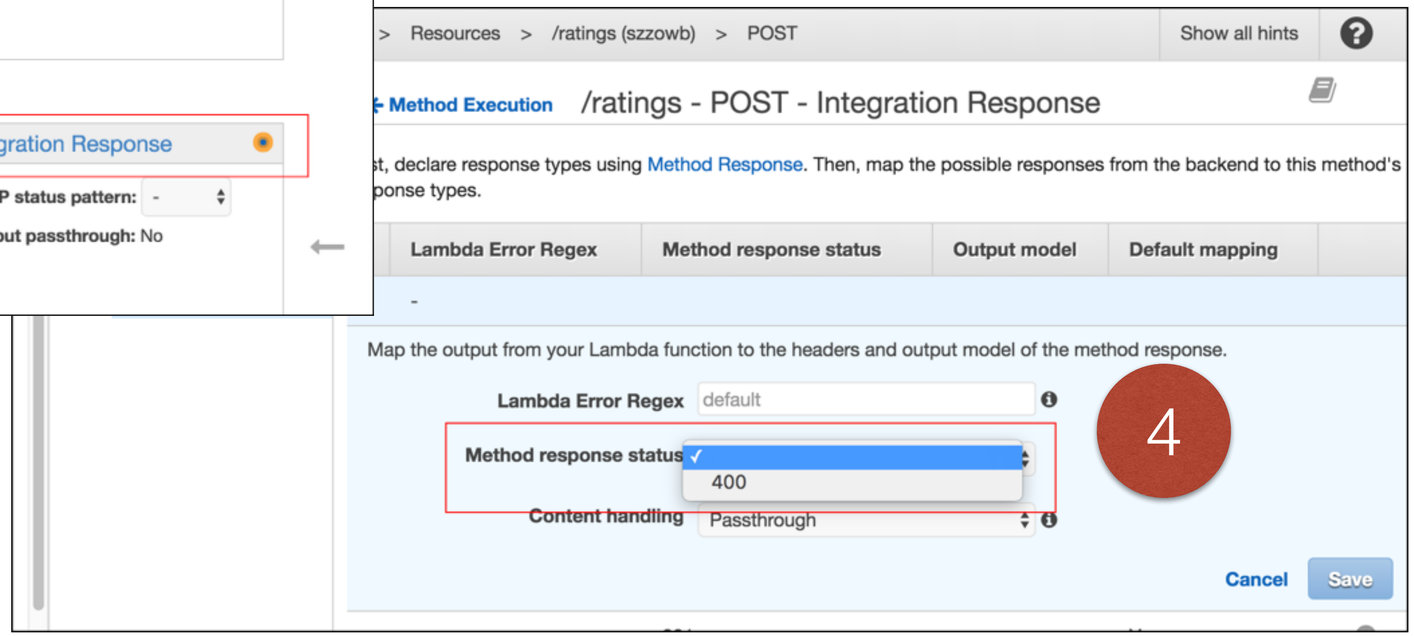
Supporting HTTP Status Codes



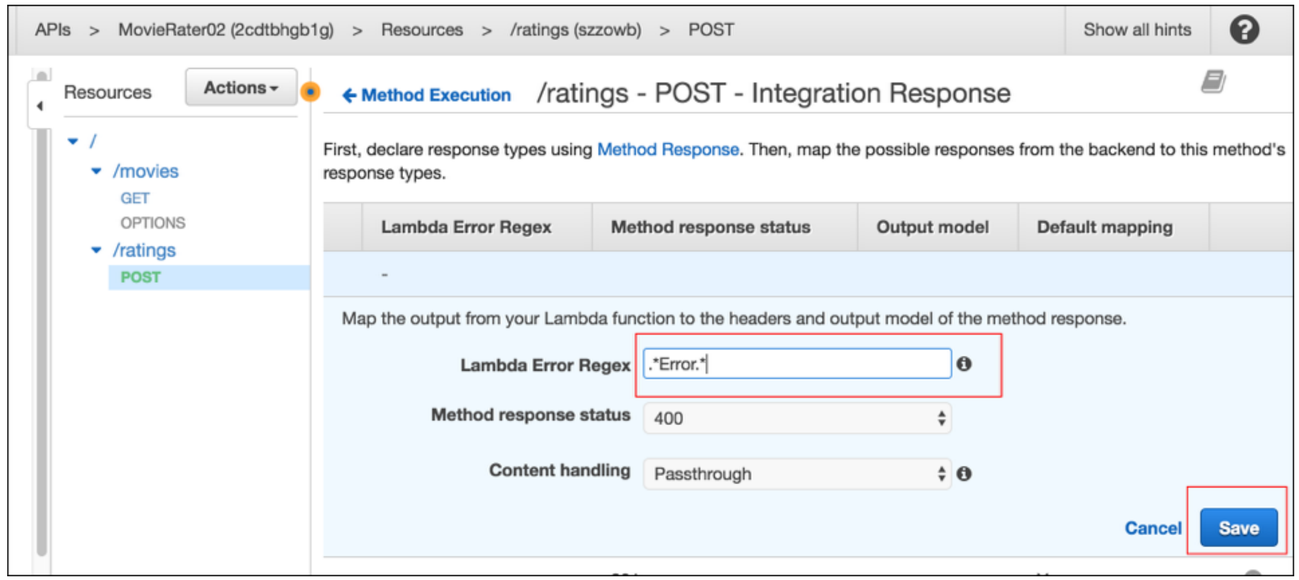


Supporting HTTP Status Codes

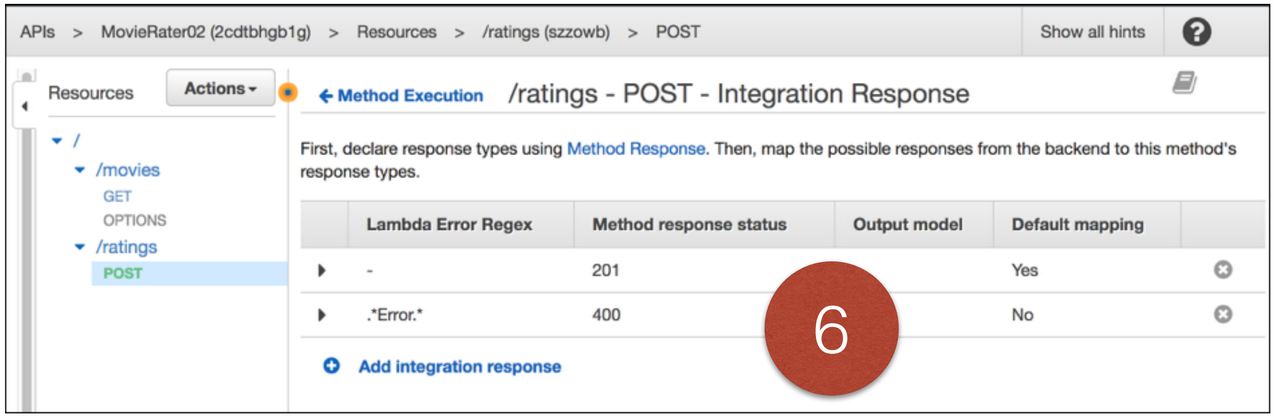




Supporting HTTP Status Codes

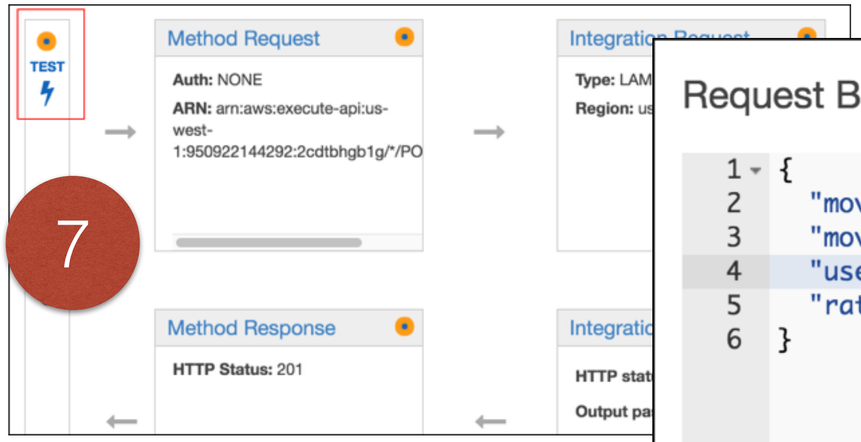


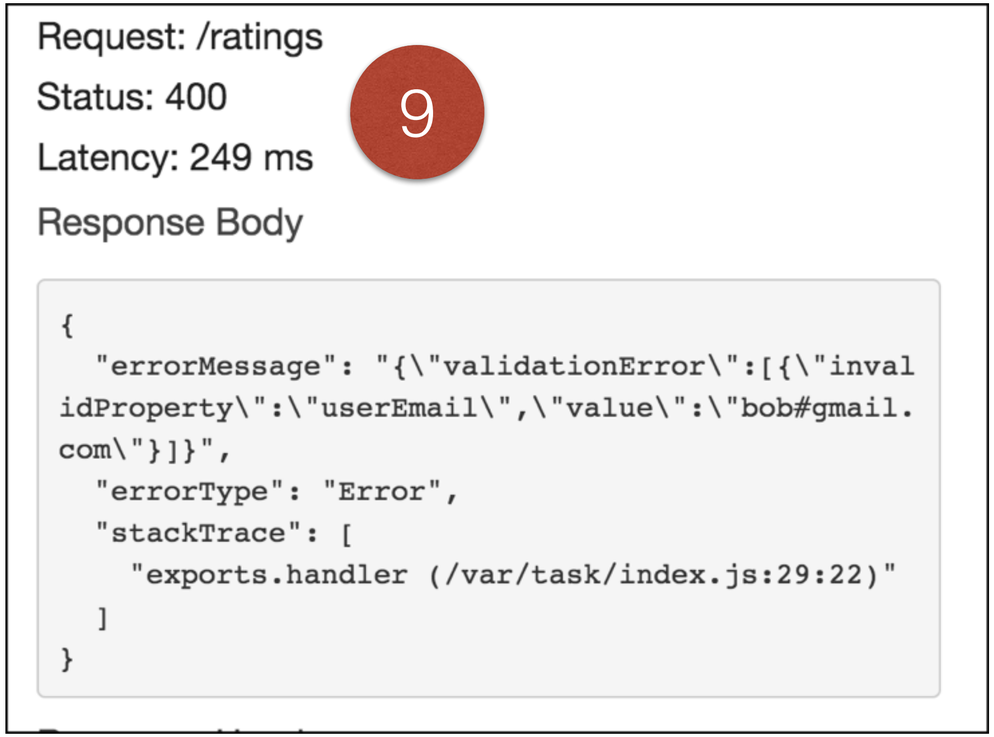
5



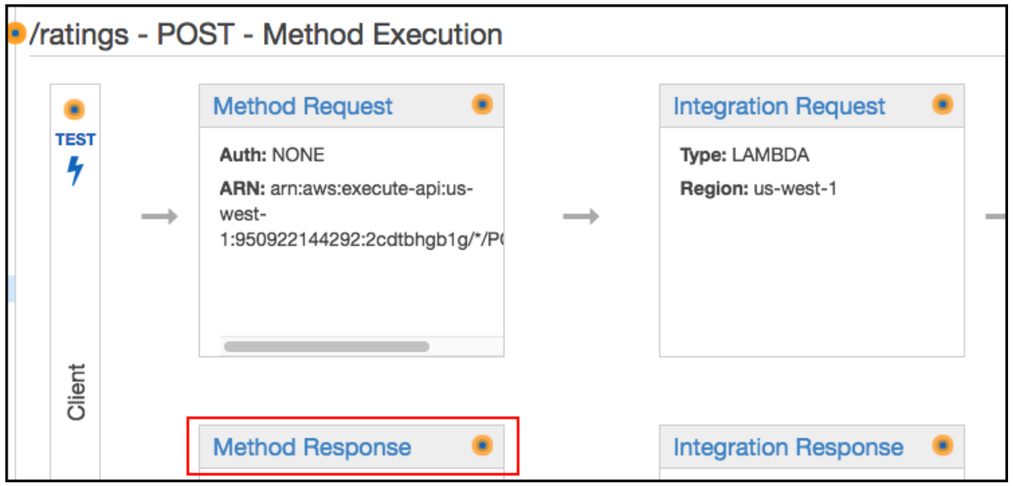
6

Supporting HTTP Status Codes



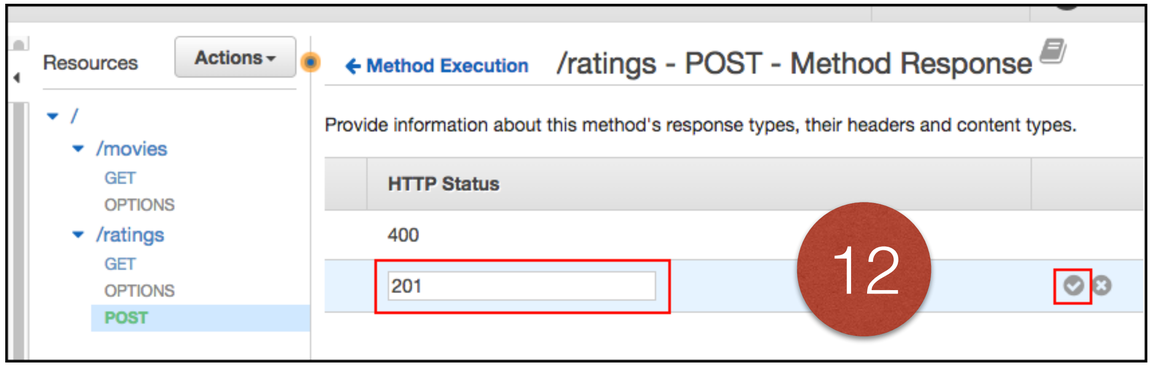


Supporting HTTP Status Codes



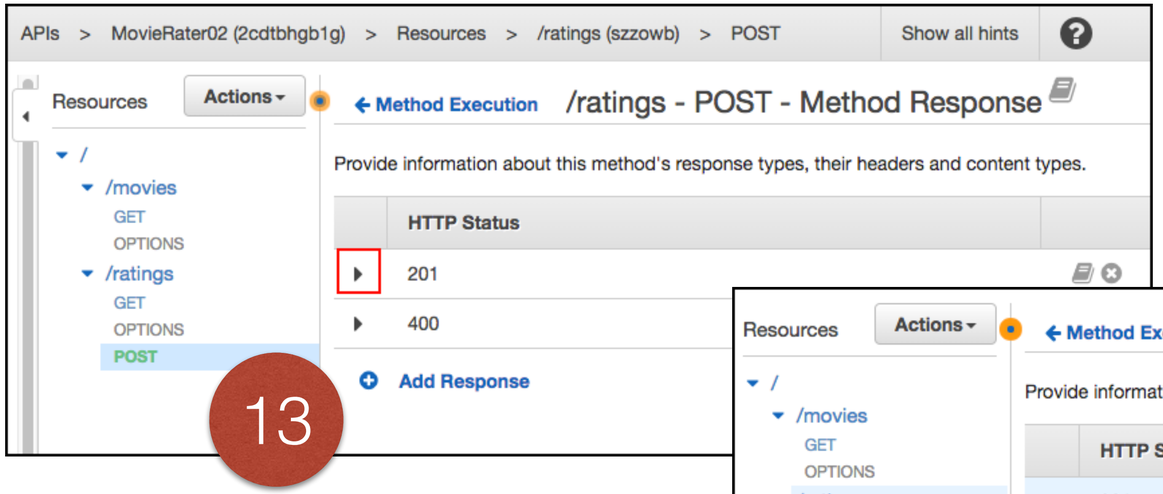
10

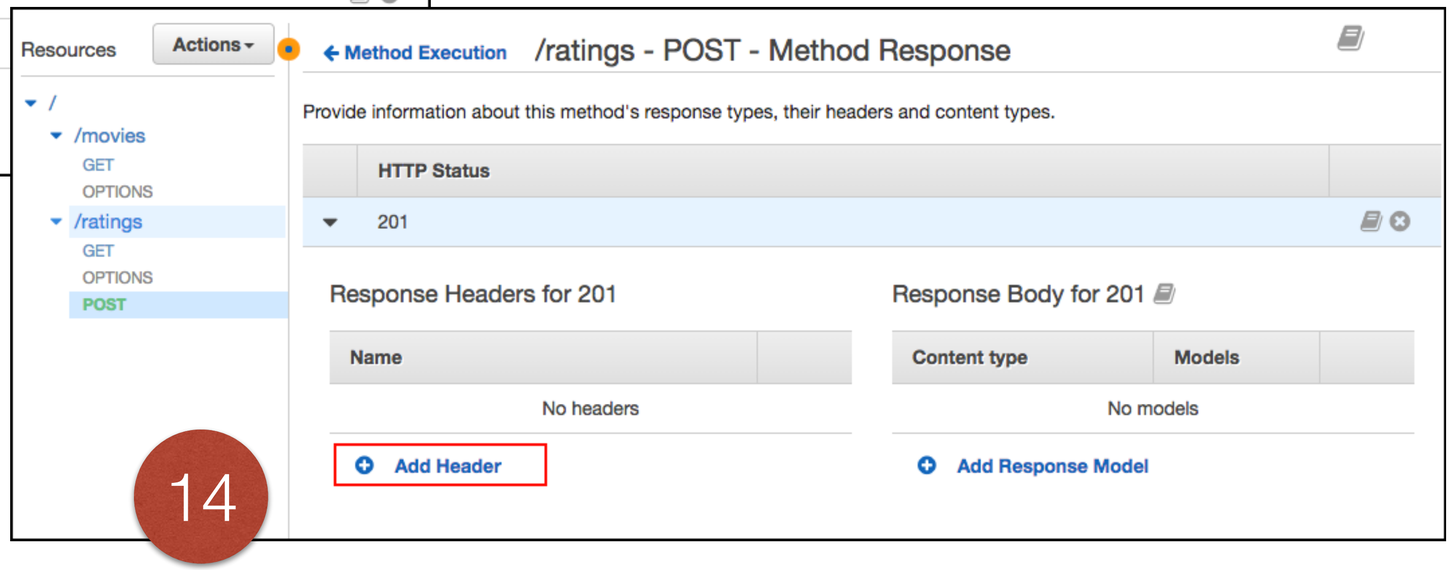
11



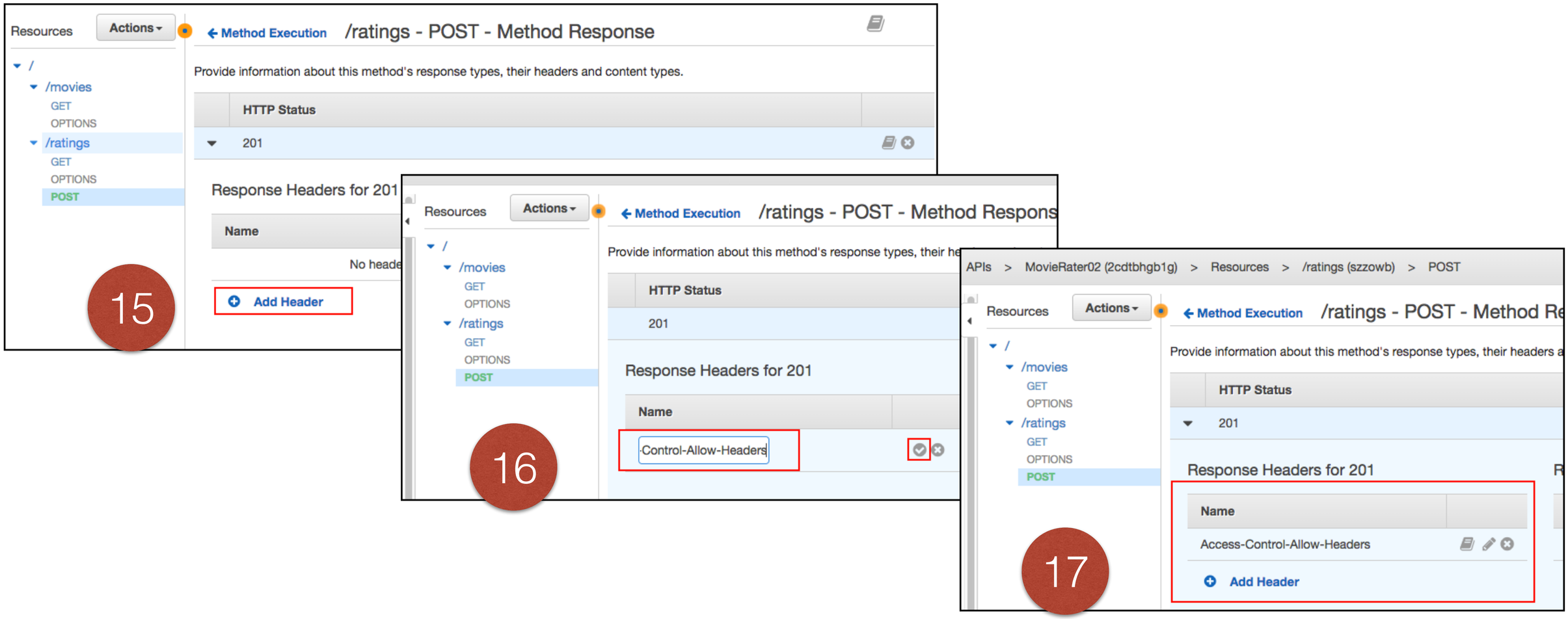
12

Supporting HTTP Status Codes

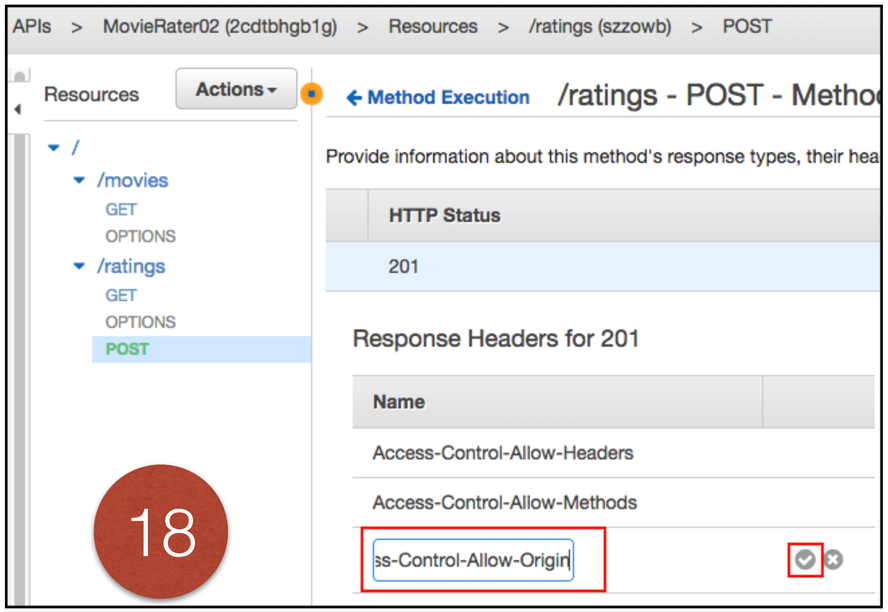


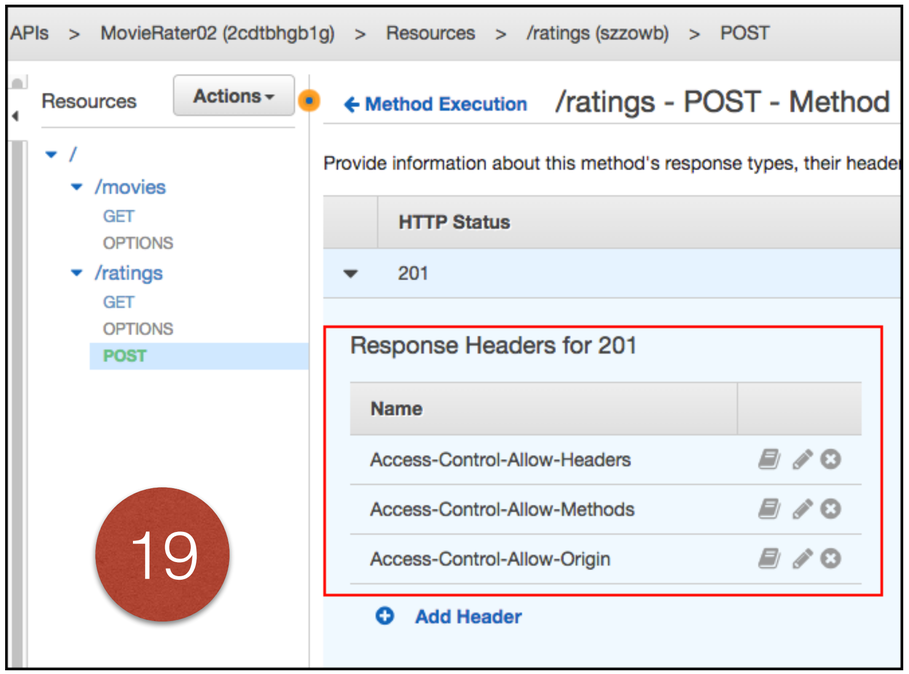


Supporting HTTP Status Codes

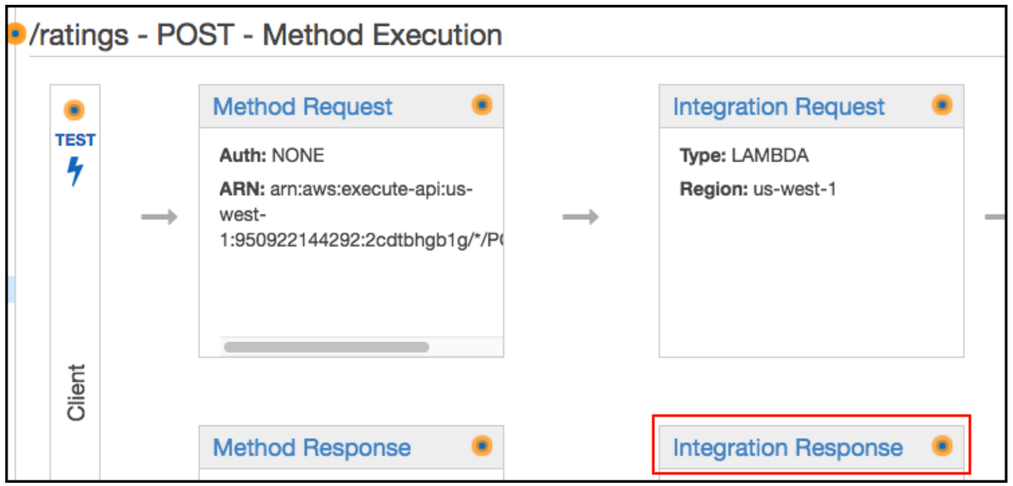


Supporting HTTP Status Codes

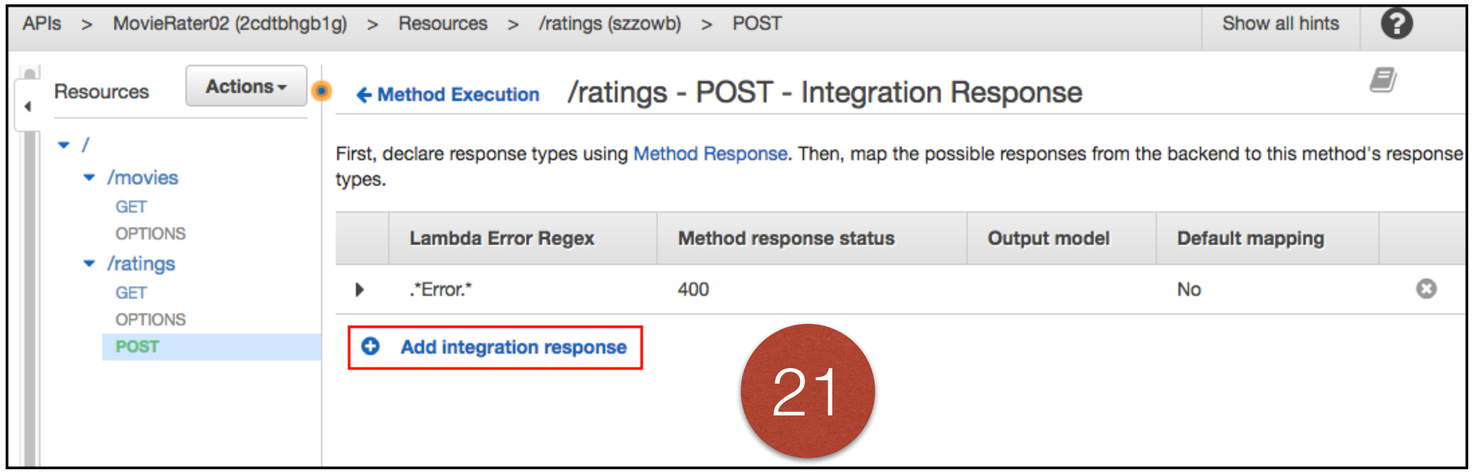




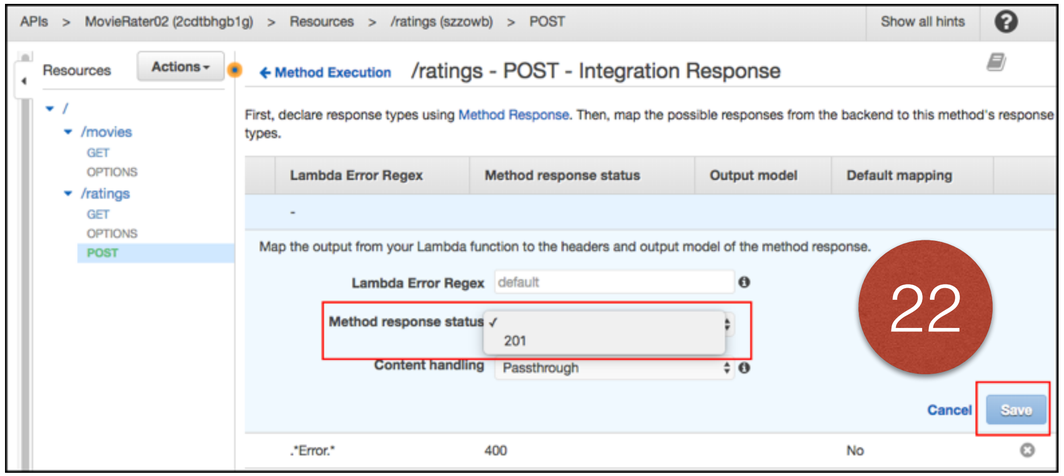
Supporting HTTP Status Codes



20

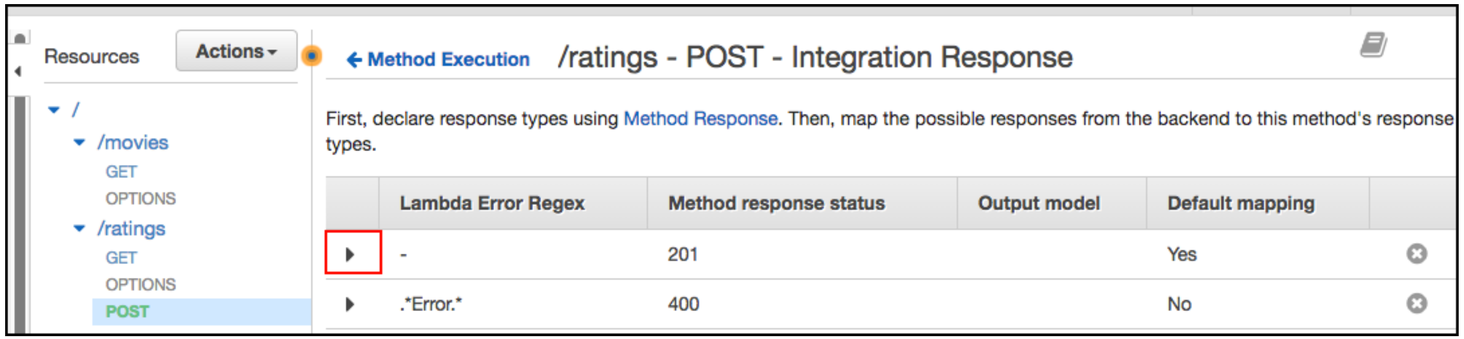


21

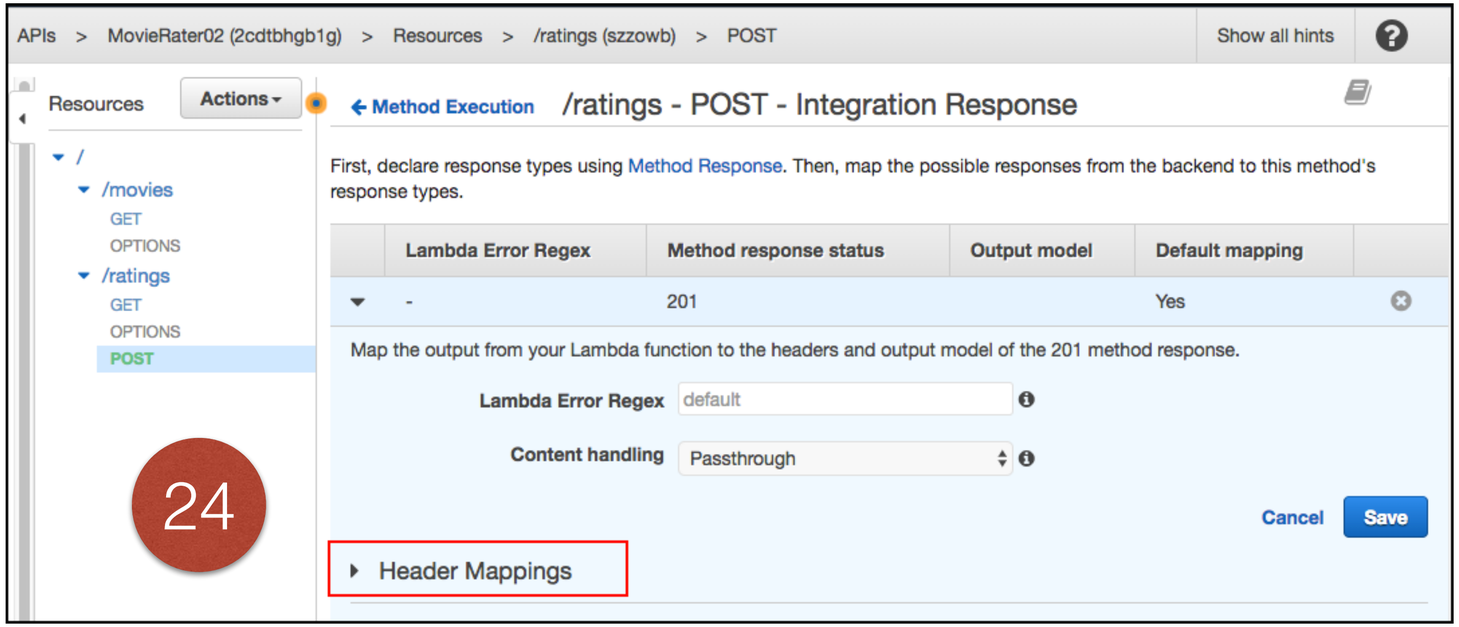


22

Supporting HTTP Status Codes

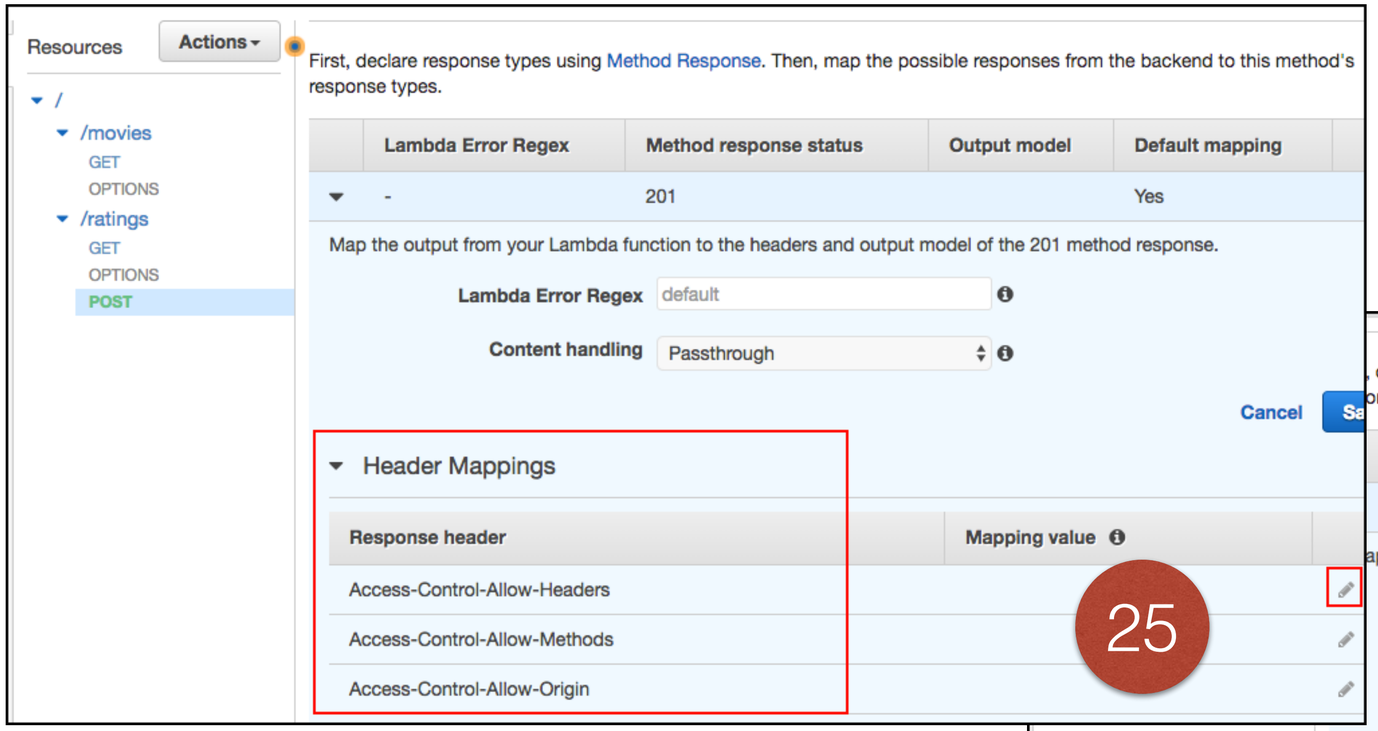


23

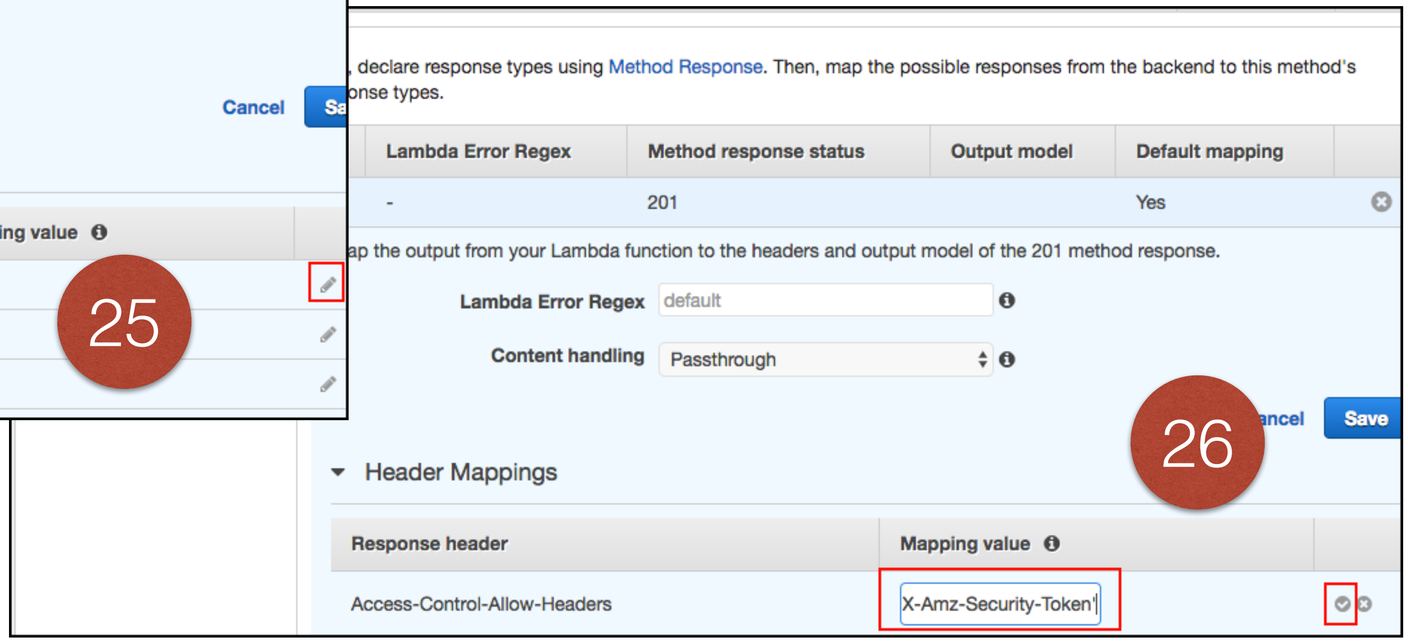


24

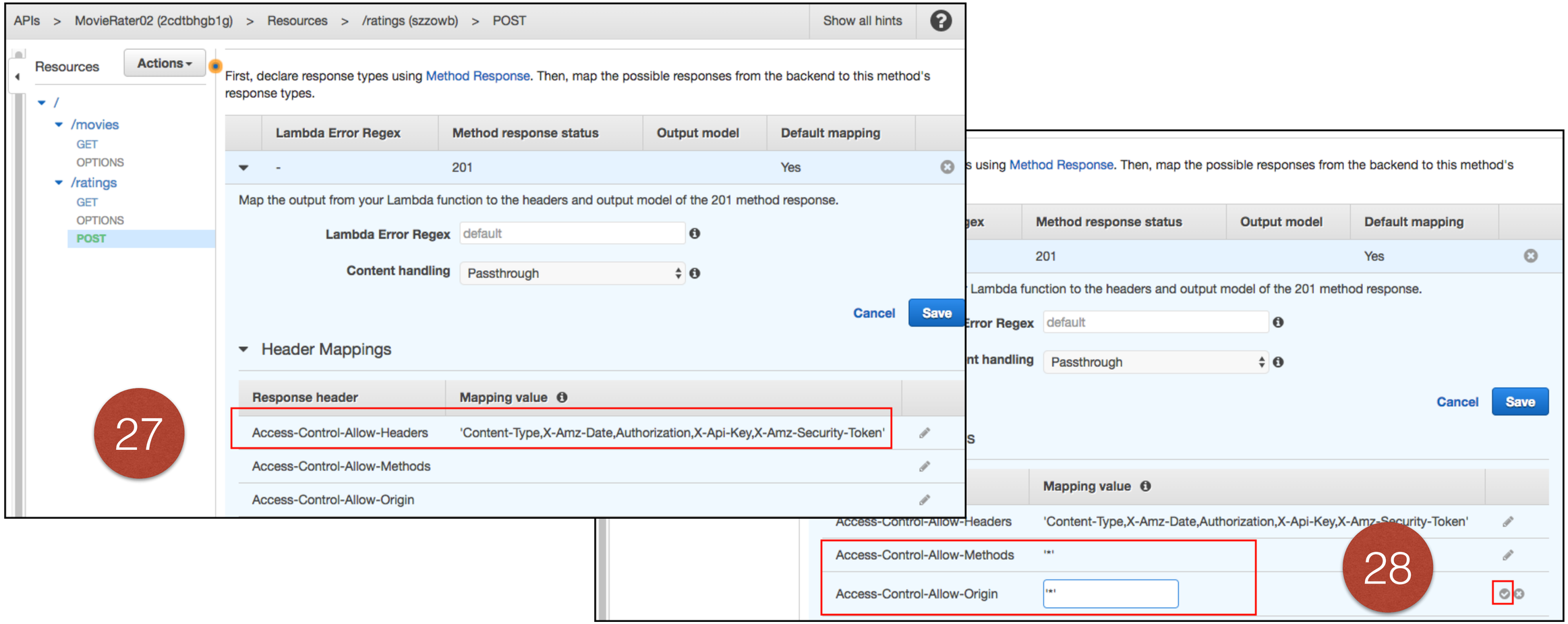
Supporting HTTP Status Codes



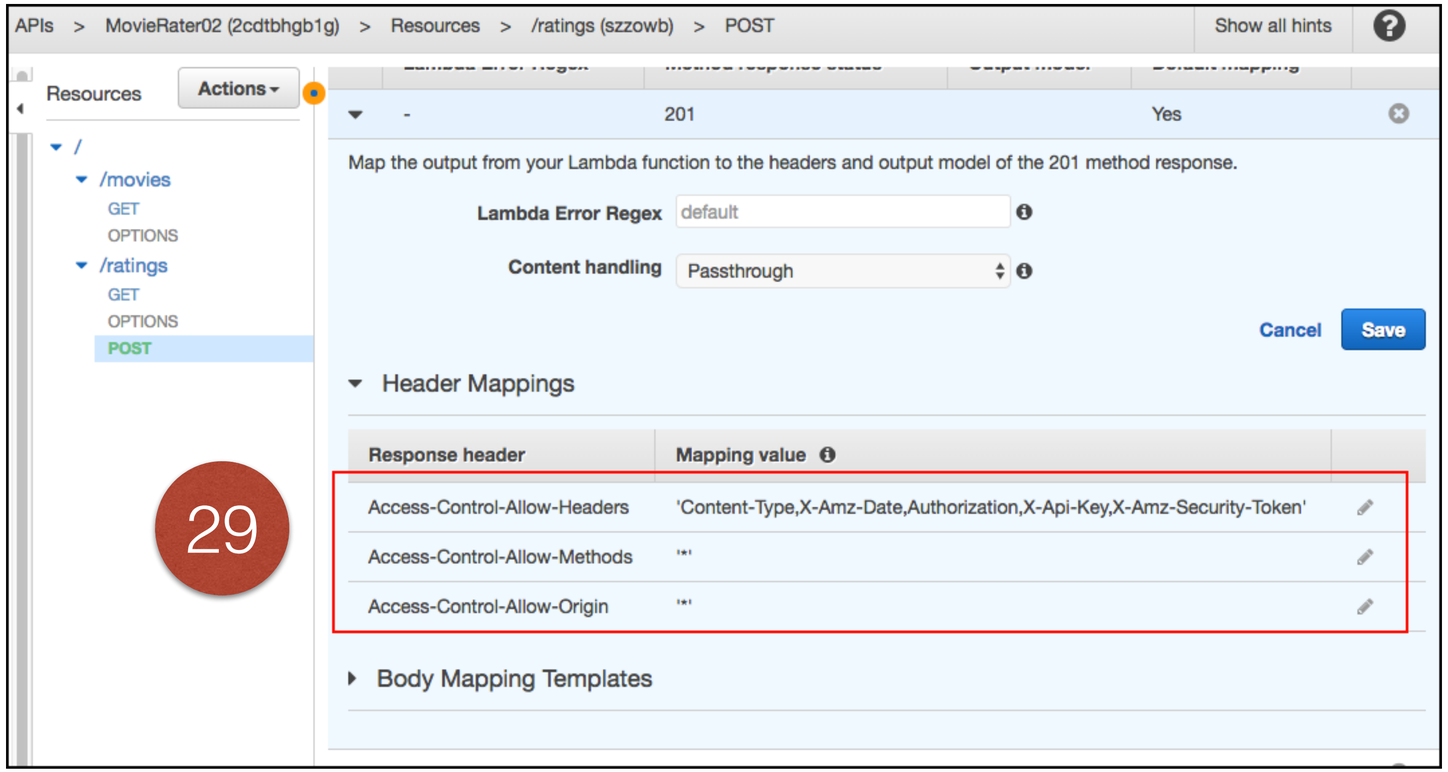
'Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token'



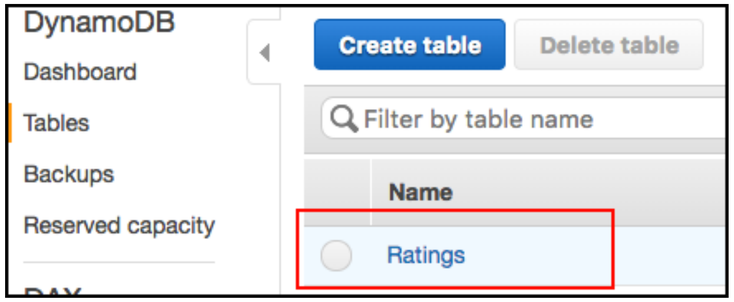
Supporting HTTP Status Codes

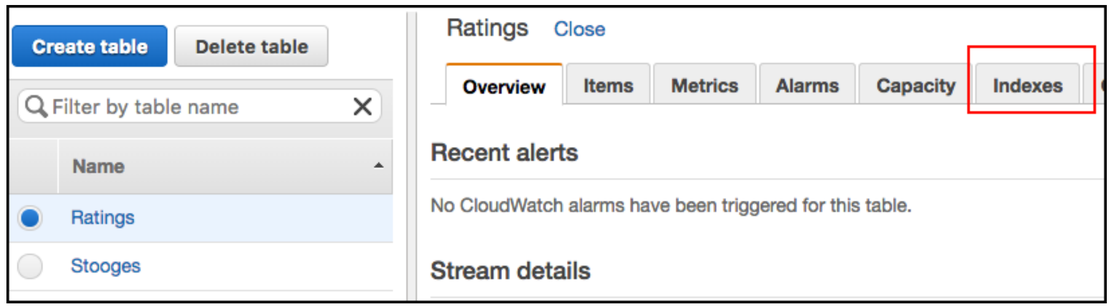


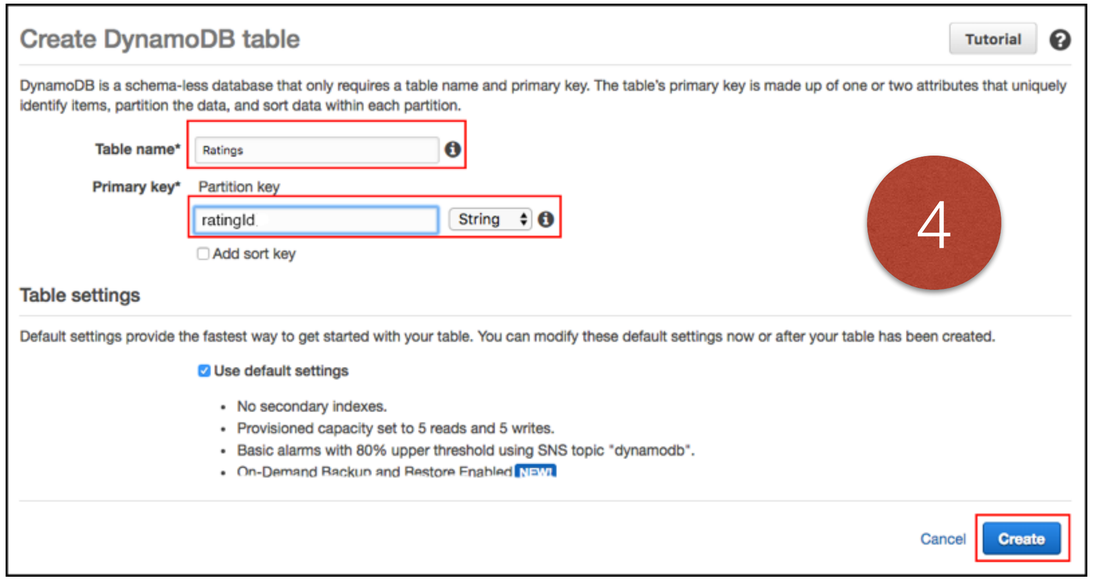
Supporting HTTP Status Codes



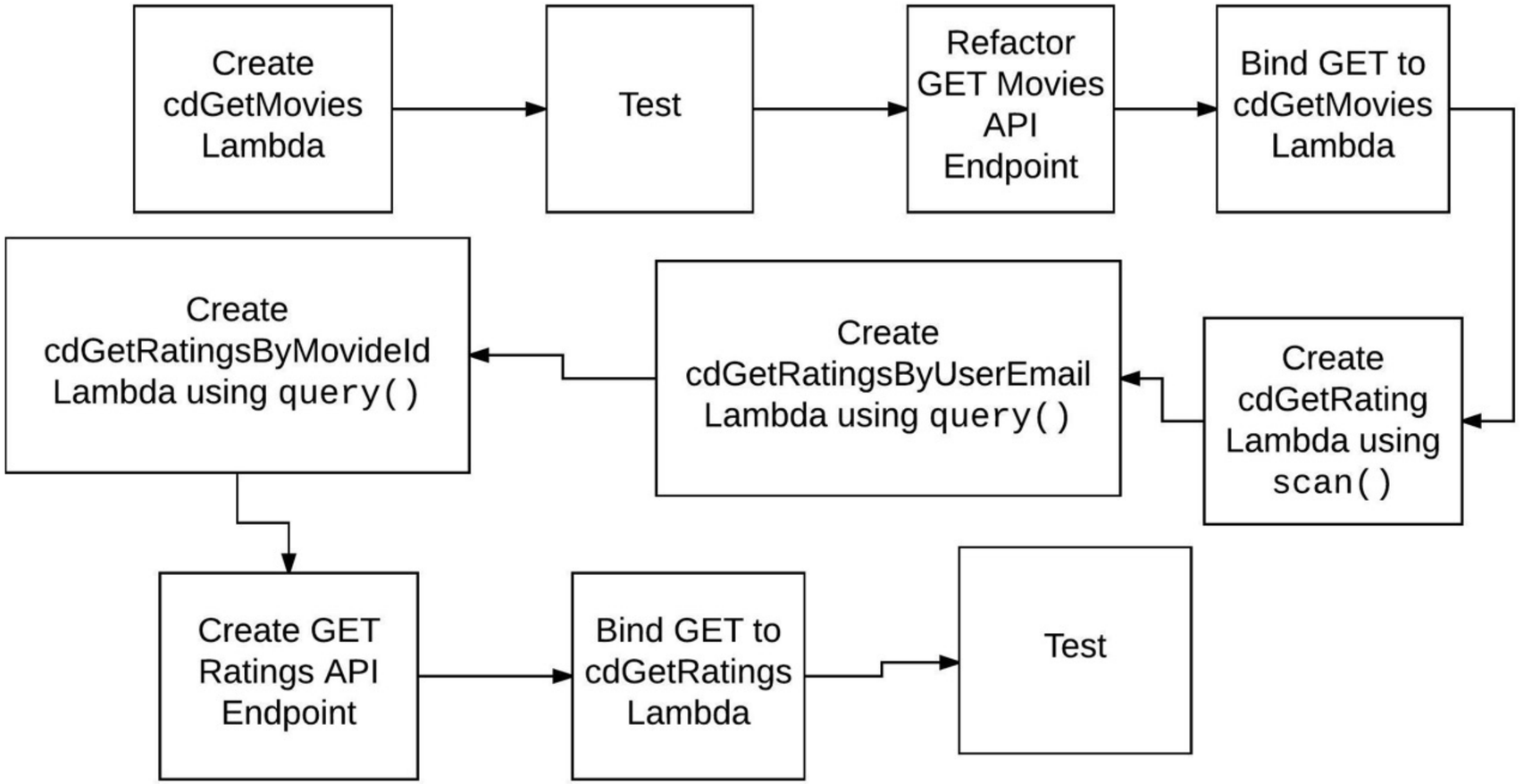
Create DynamoDB Index



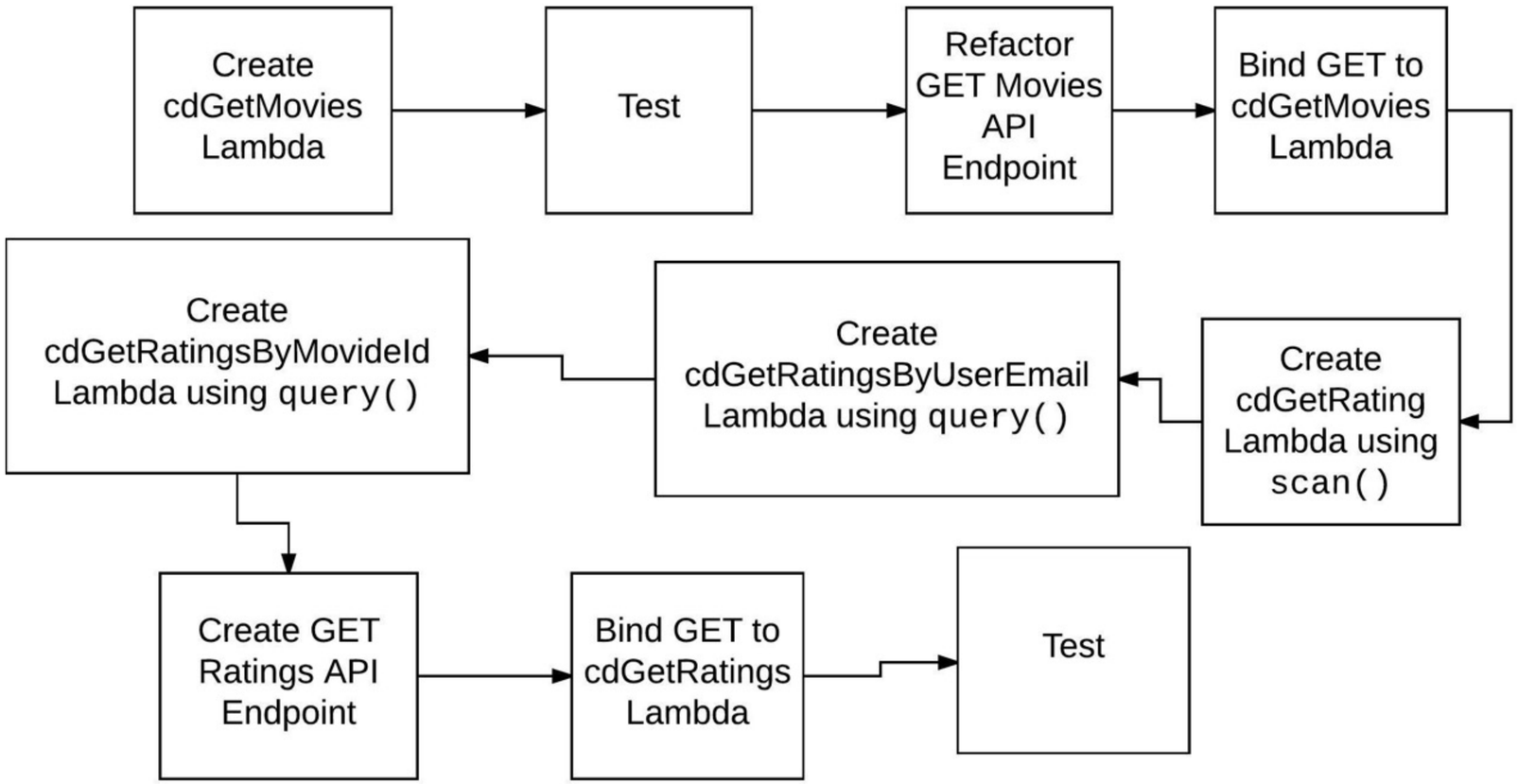




Create DynamoDB Index

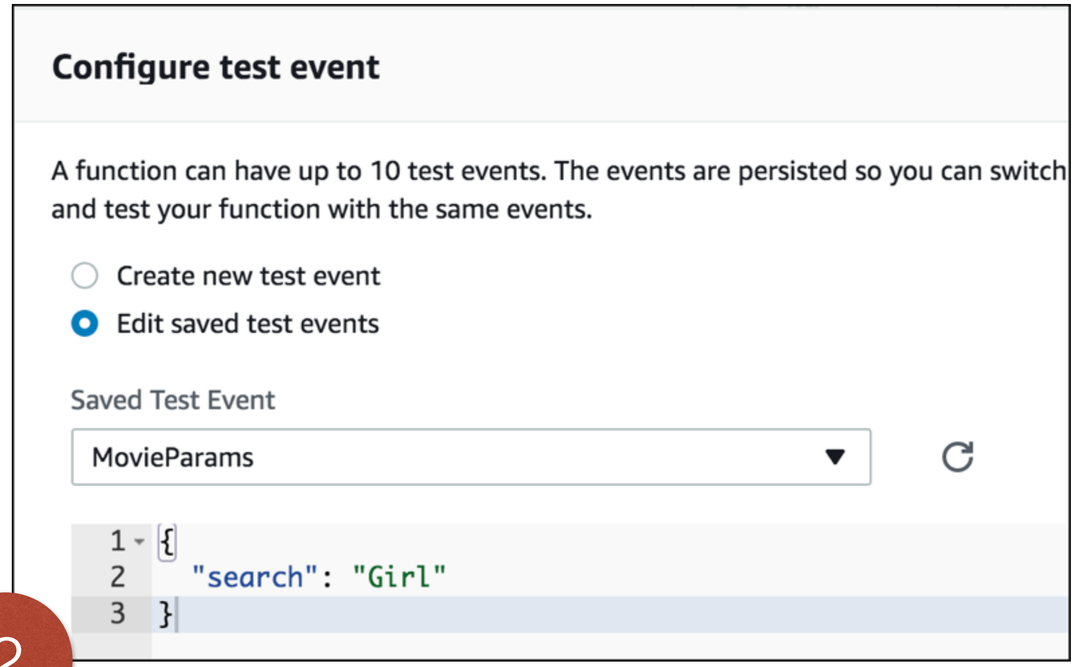


Session 5

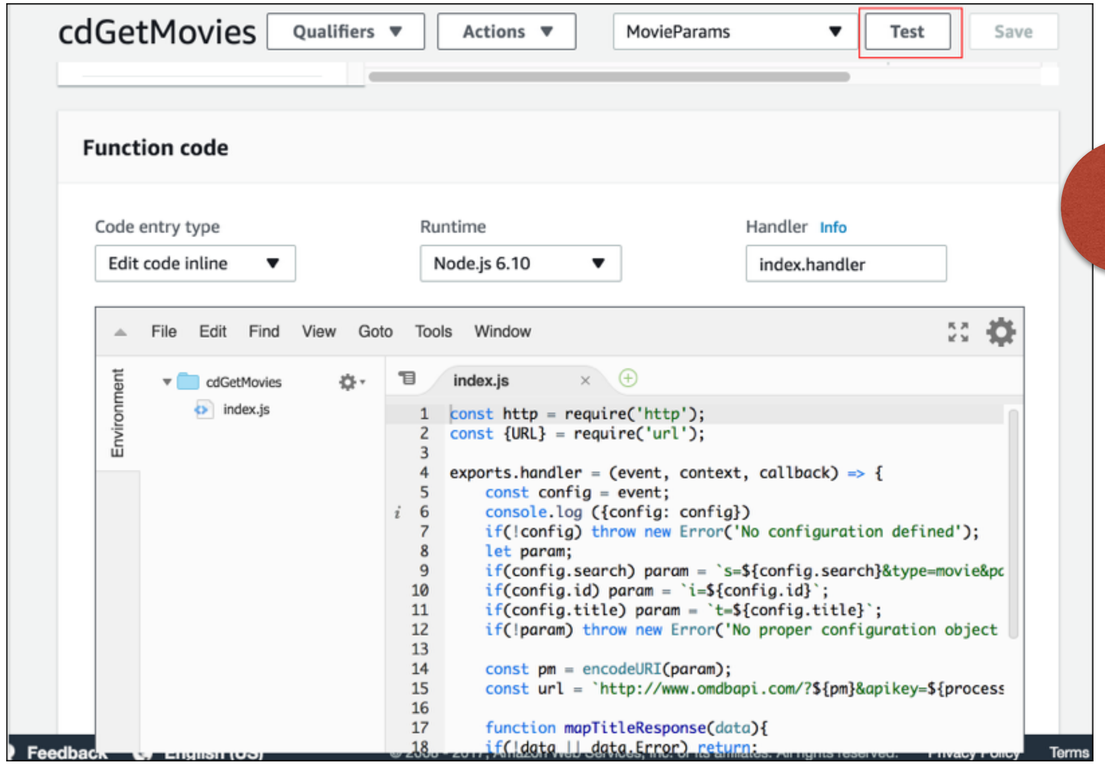

Create cdGetMovies Lambda
Test
Refactor GET Movies API Endpoint
Bind GET to cdGetMovies Lambda
Create cdGetRatingsByMovideId Lambda using query()
Create cdGetRatingsByUserEmail Lambda using query()
Create cdGetRating Lambda using scan()
Create GET Ratings API Endpoint
Bind GET to cdGetRatings Lambda
Test

Create cdGetMovies Lambda

1 https://github.com/reselbob/CDAwsClass/blob/Session-5/getMovies.js



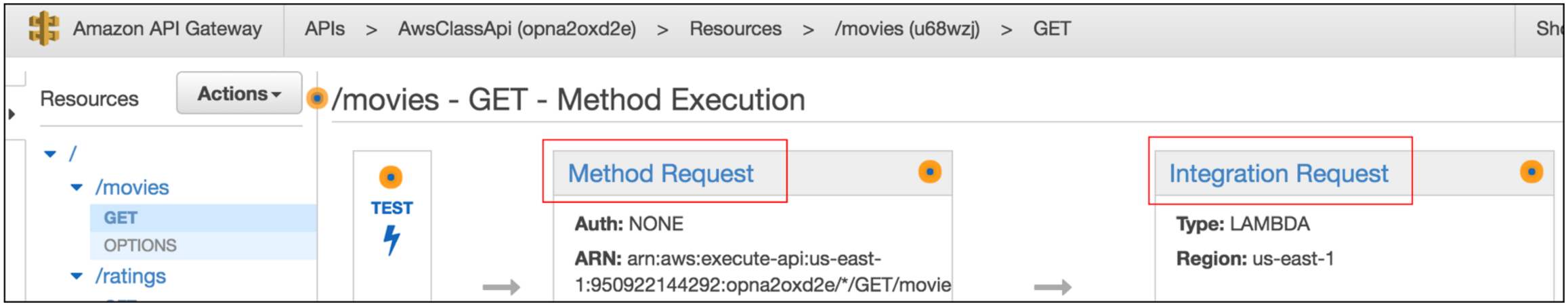
2



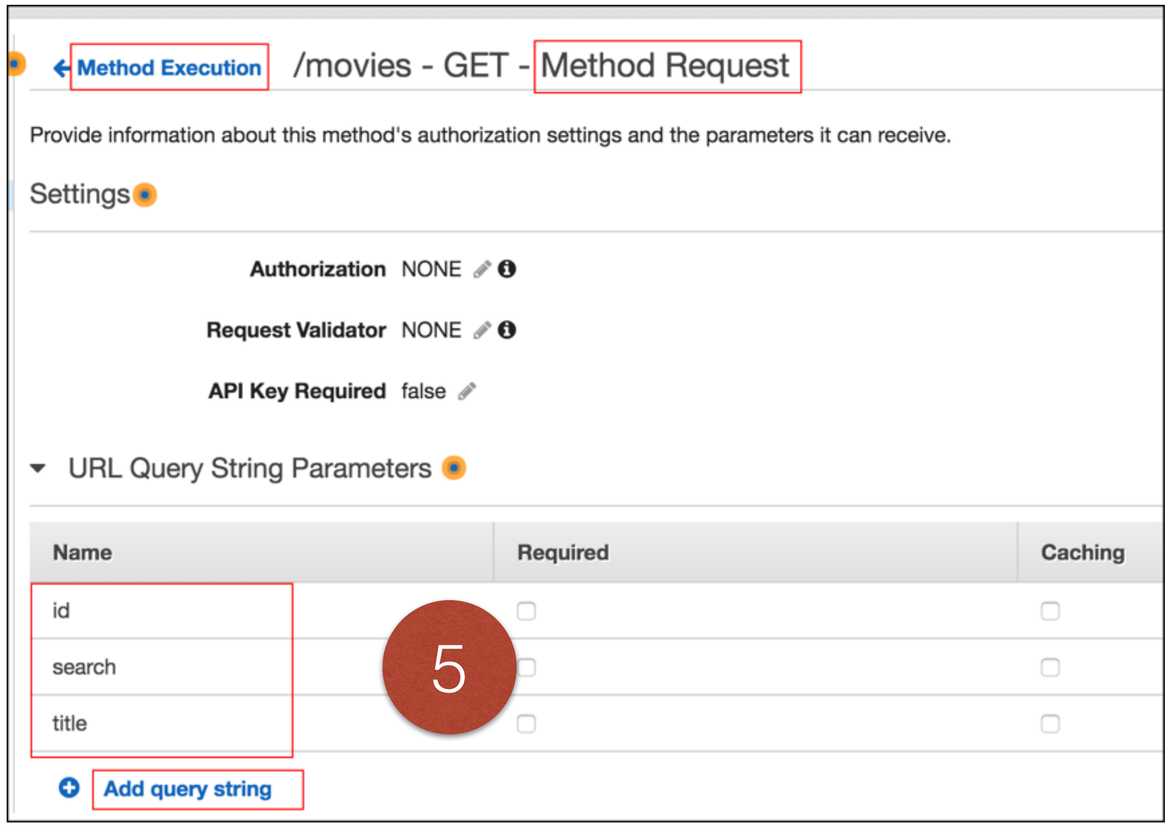
3

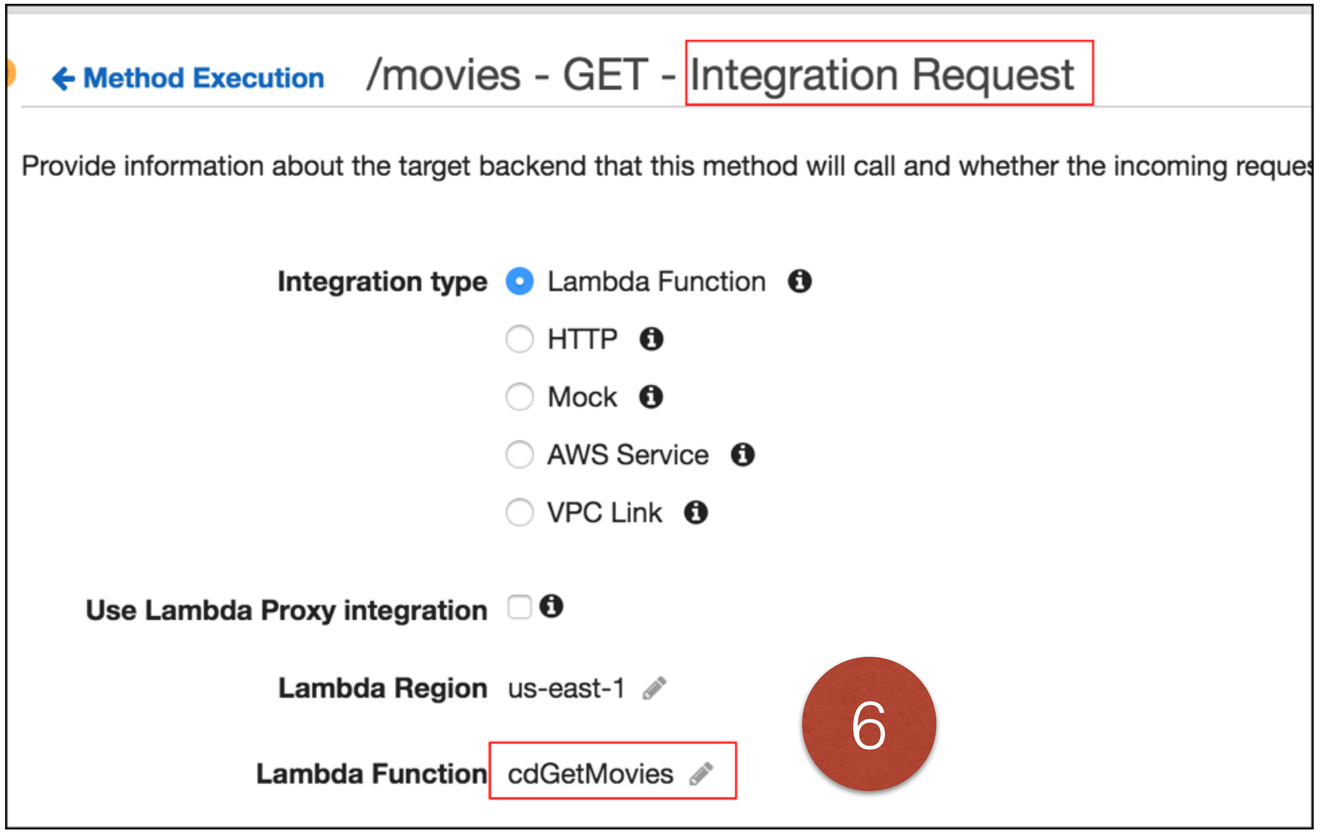
Refactor GET Movies Endpoint

4

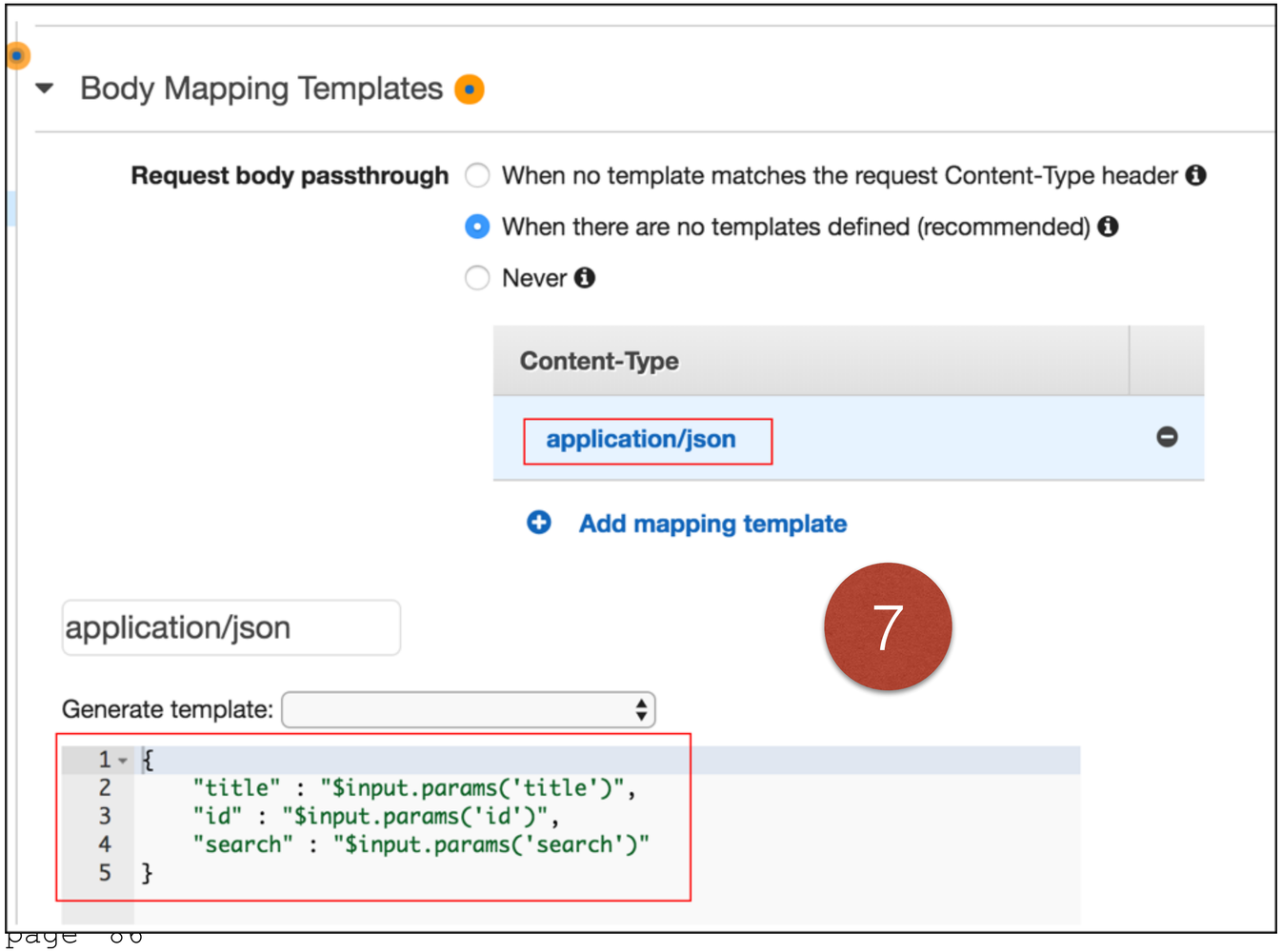


Refactor GET Movies Endpoint



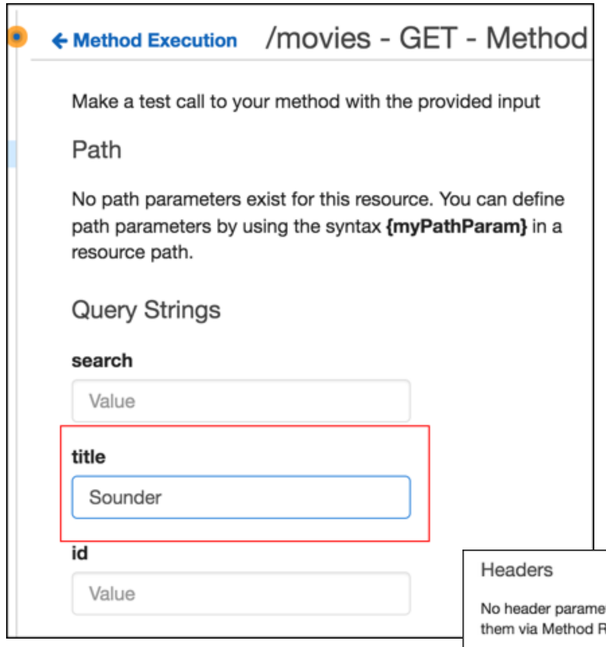


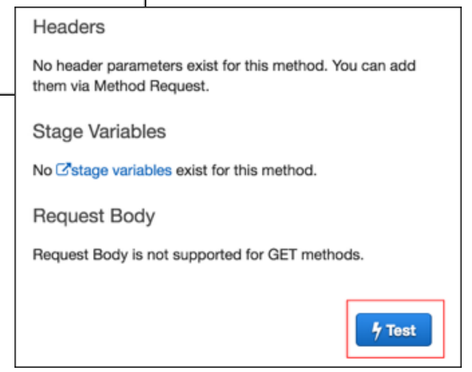
Refactor GET Movies Endpoint



7

8





9

Refactor GET Movies Endpoint

10

Request: /movies?title=Sounder

Status: 200

Latency: 105 ms

Response Body


```
{
  "genre": "",
  "id": "tt0069303",
  "title": "Sounder",
  "releaseDate": "10 May 1973",
  "studio": "Rainbow Group / KOCH Entertainment"
}
```


Response Headers


```
{"X-Amzn-Trace-Id":"sampled=0;root=1-5a2cafcb-95e58a3f7d7c142c
b851ba64","Content-Type":"application/json"}
```

Created getRatings Lambda

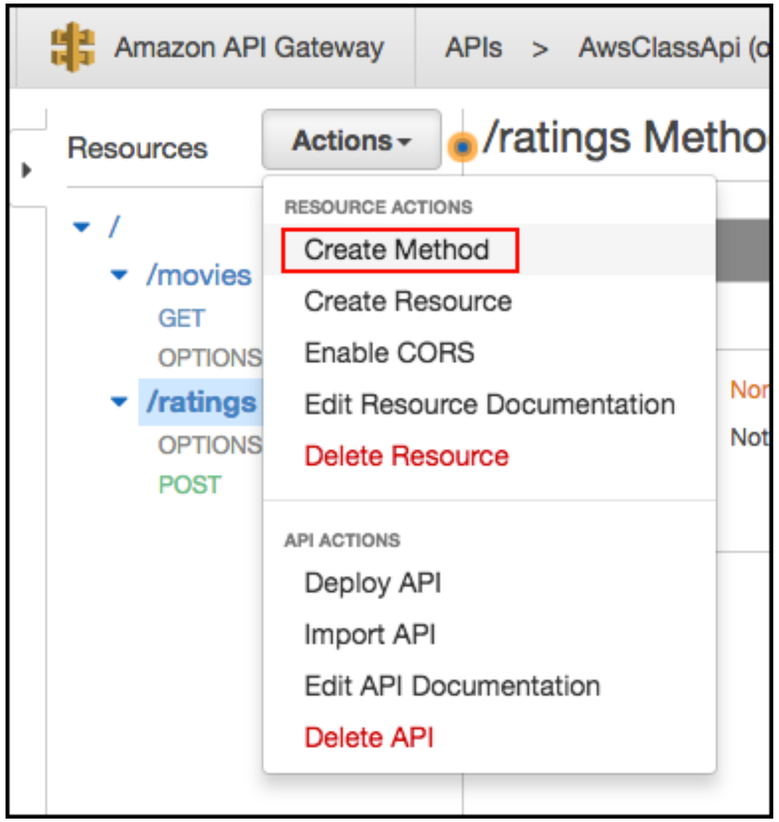
1. http://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/DynamoDB.html#query-property

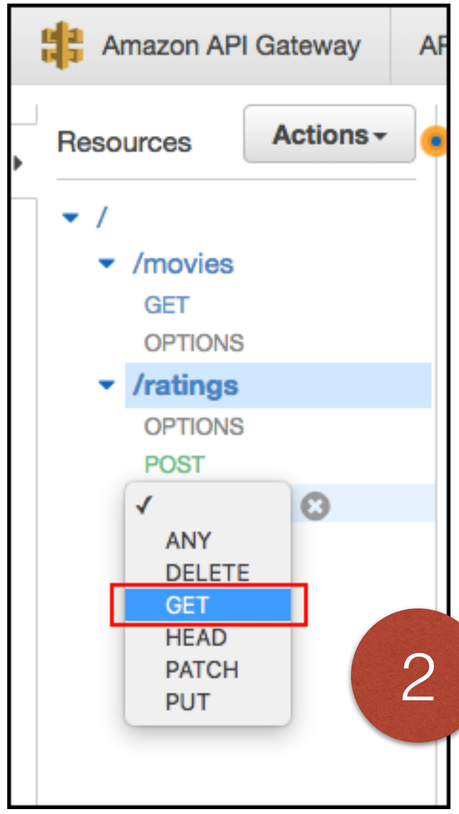
2. https://github.com/reselbob/CDAwsClass/blob/Session-5/getRatingsByMovieId.js

3. https://github.com/reselbob/CDAwsClass/blob/Session-5/getRatingsByUserEmail.js

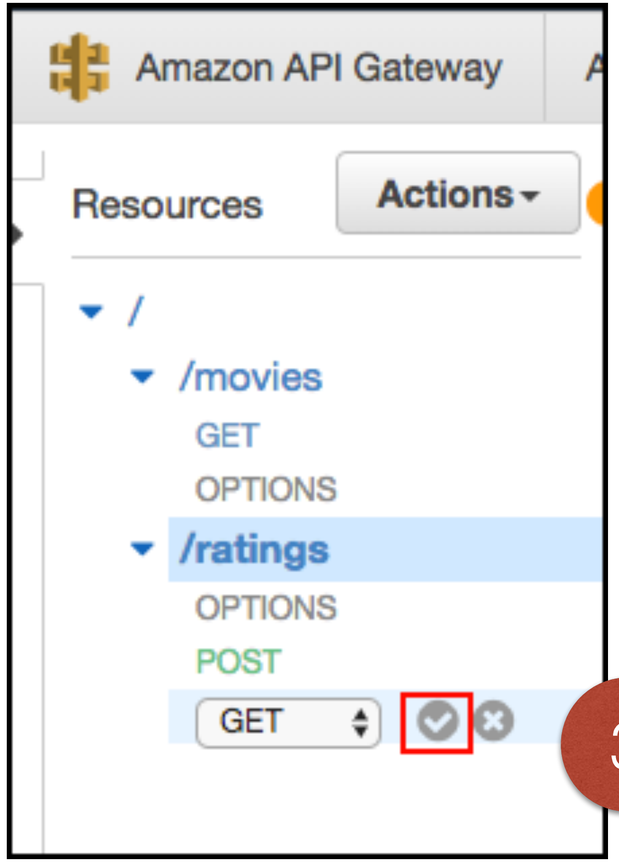
Create the GET Ratings Endpoint

1



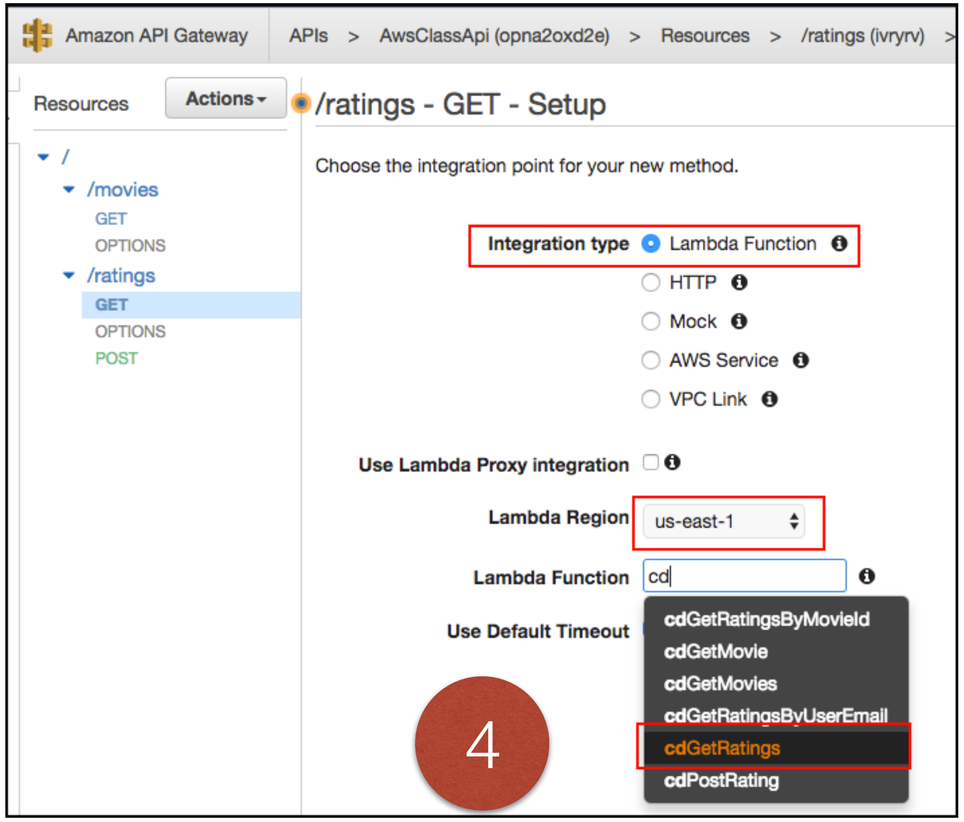


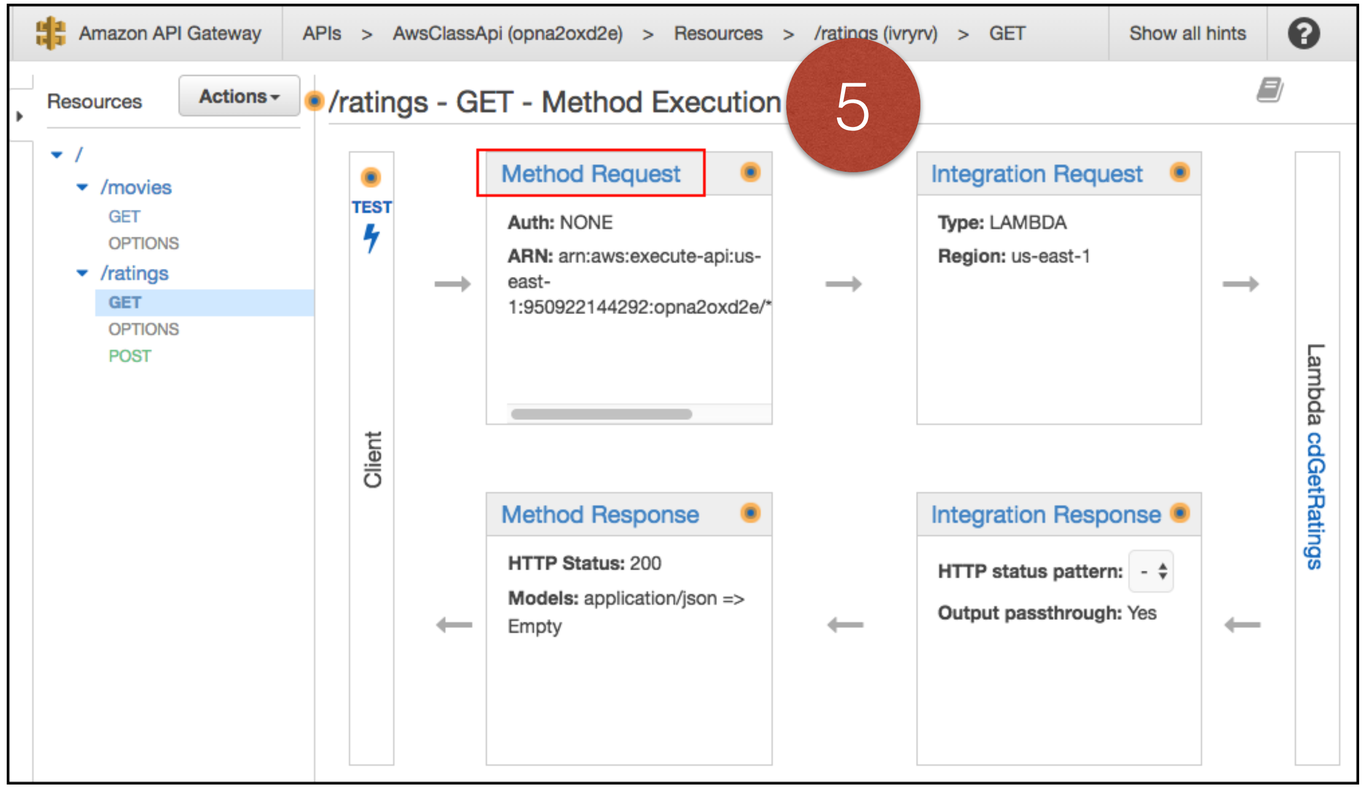
2



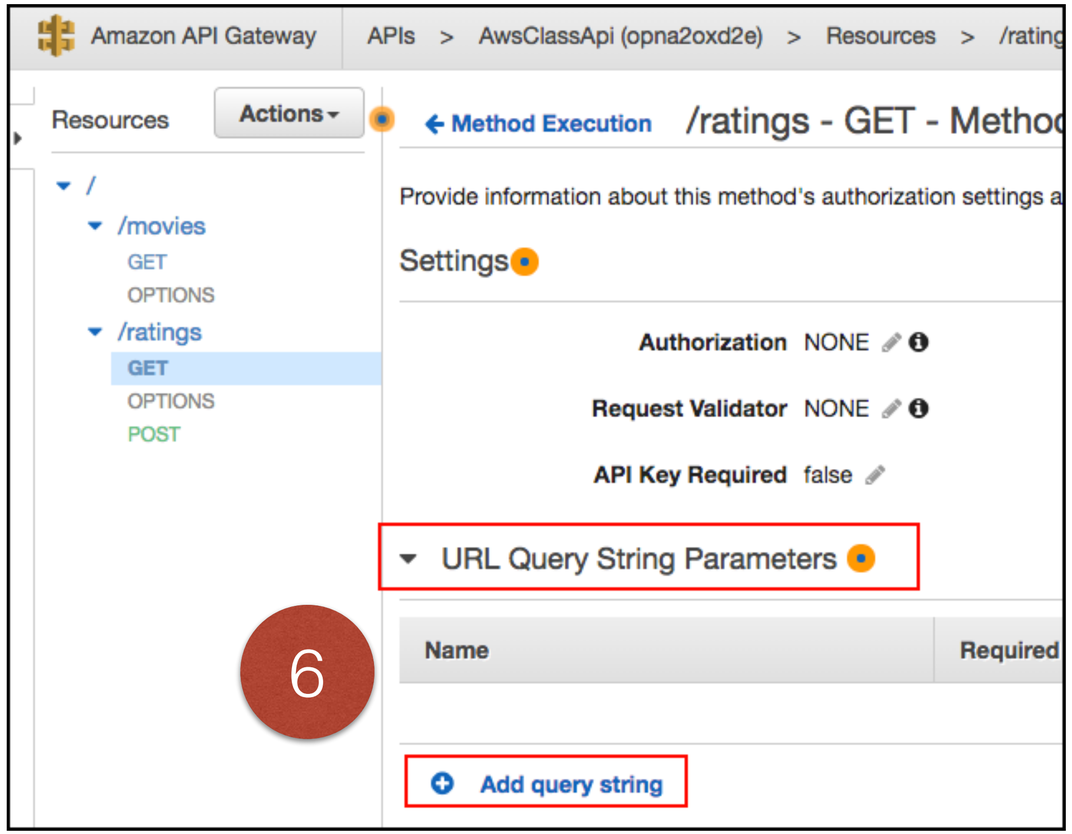
3

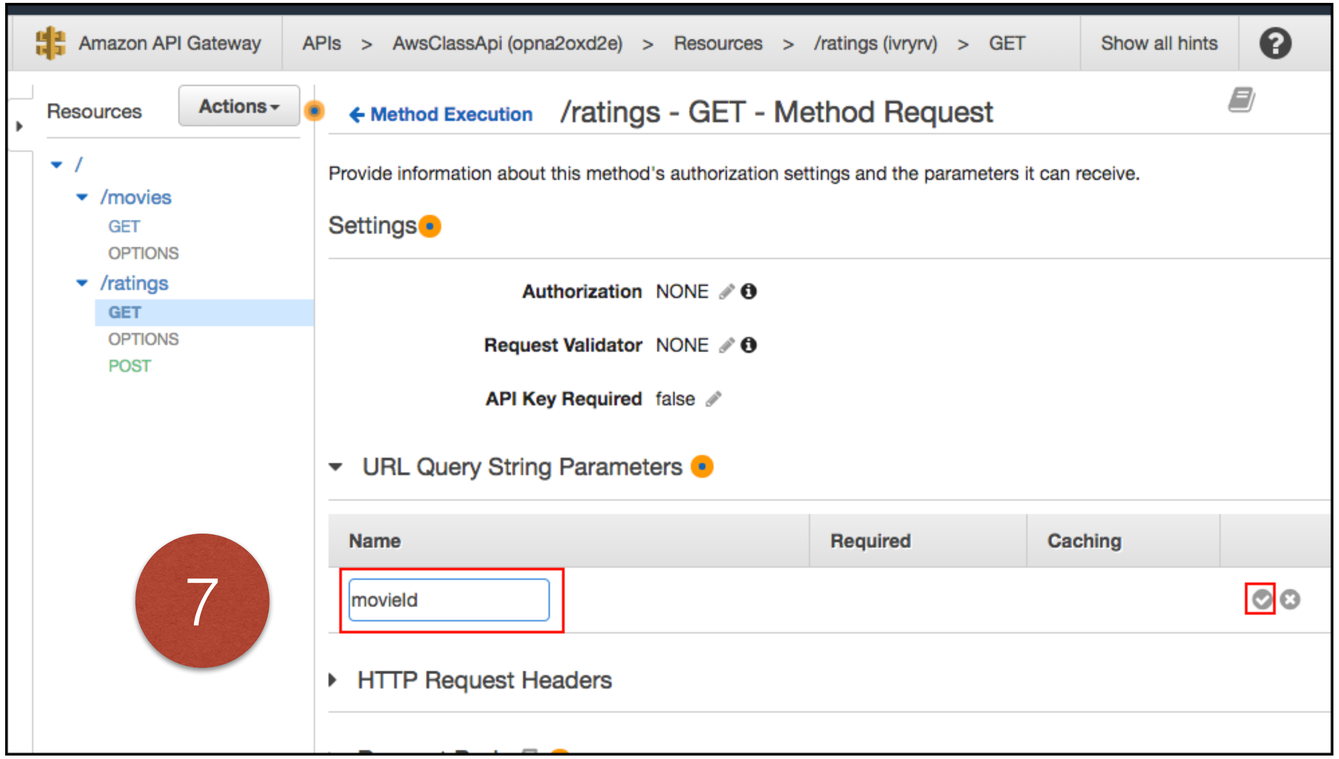
Create the GET Ratings Endpoint



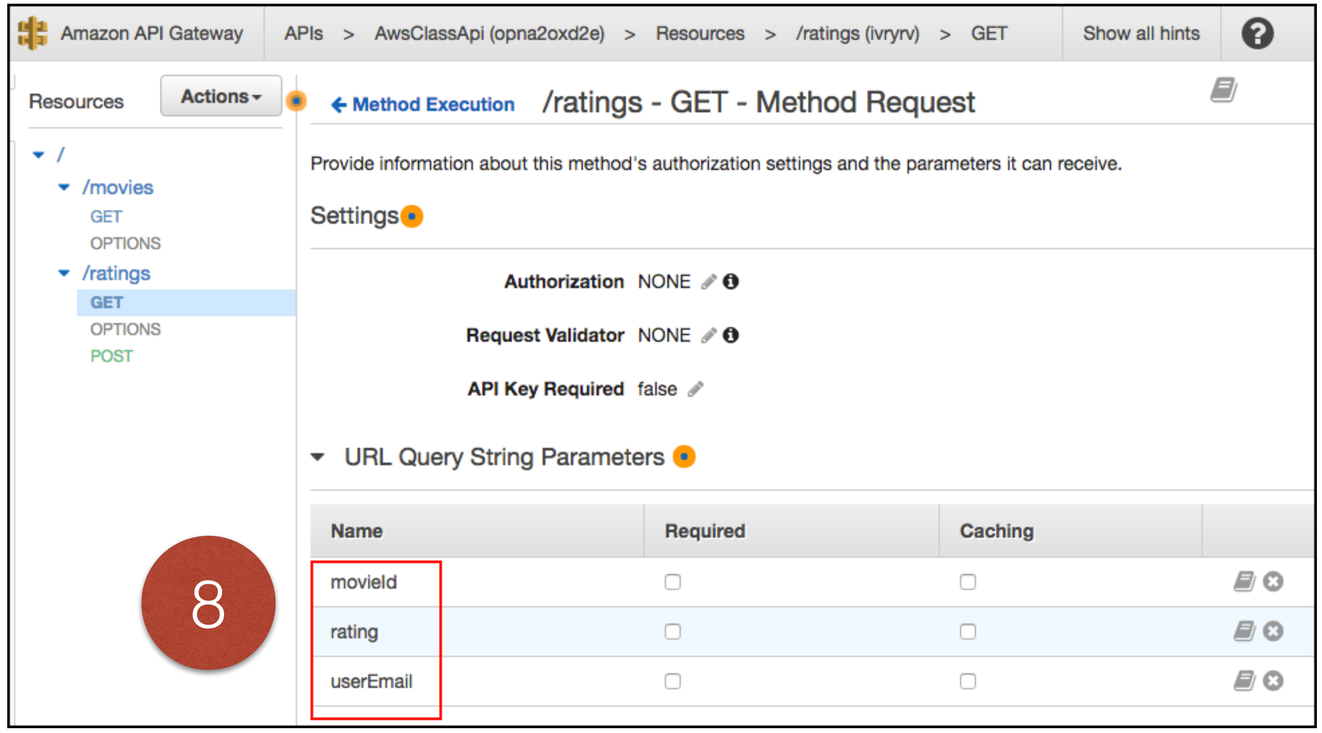


Create the GET Ratings Endpoint

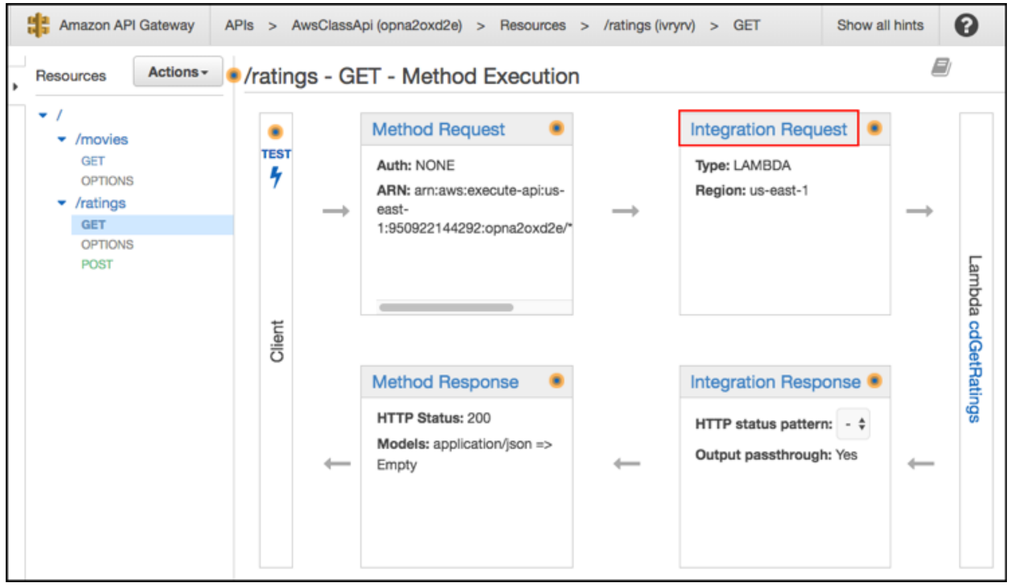




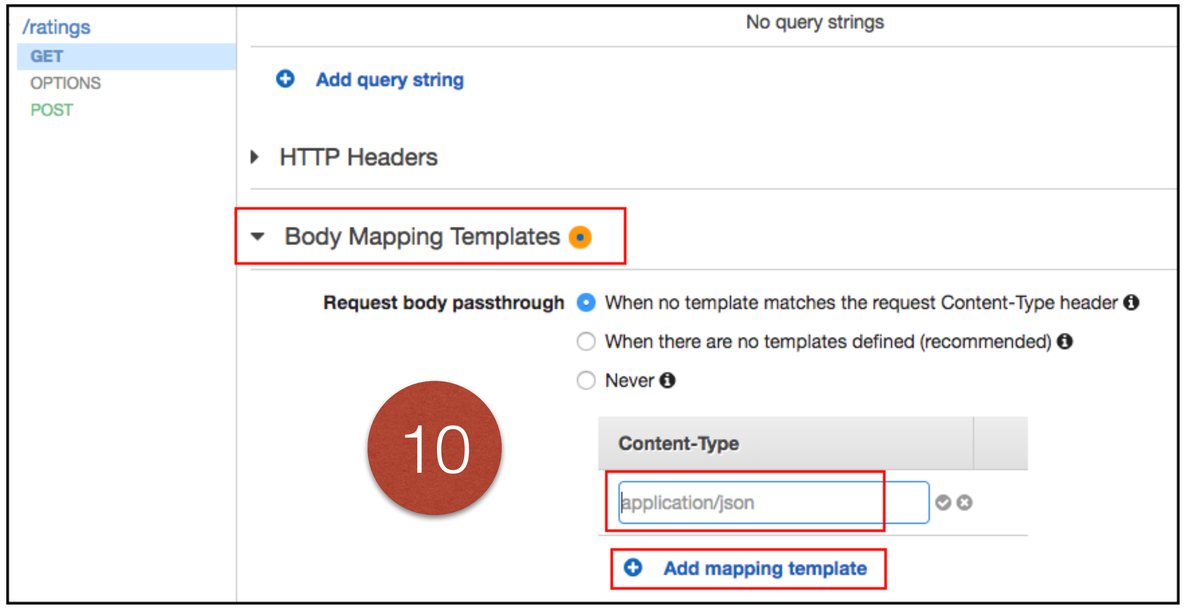
Create the GET Ratings Endpoint

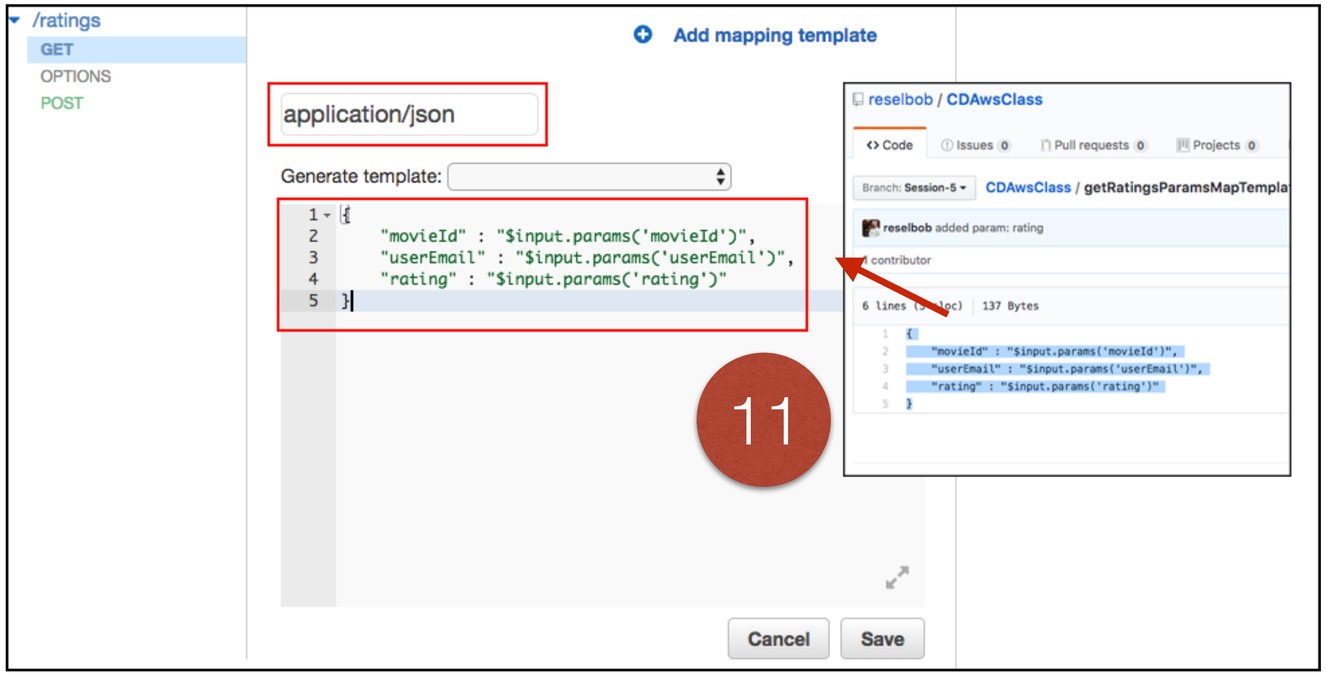


9



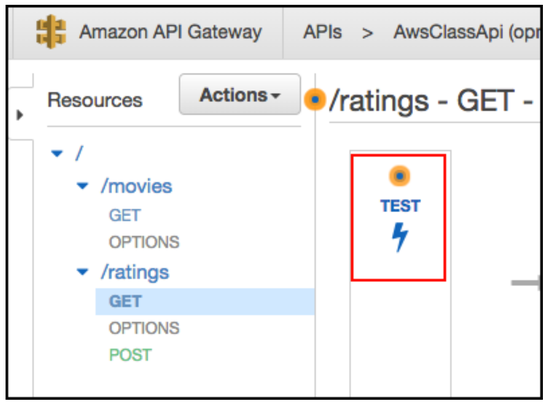
Create the GET Ratings Endpoint

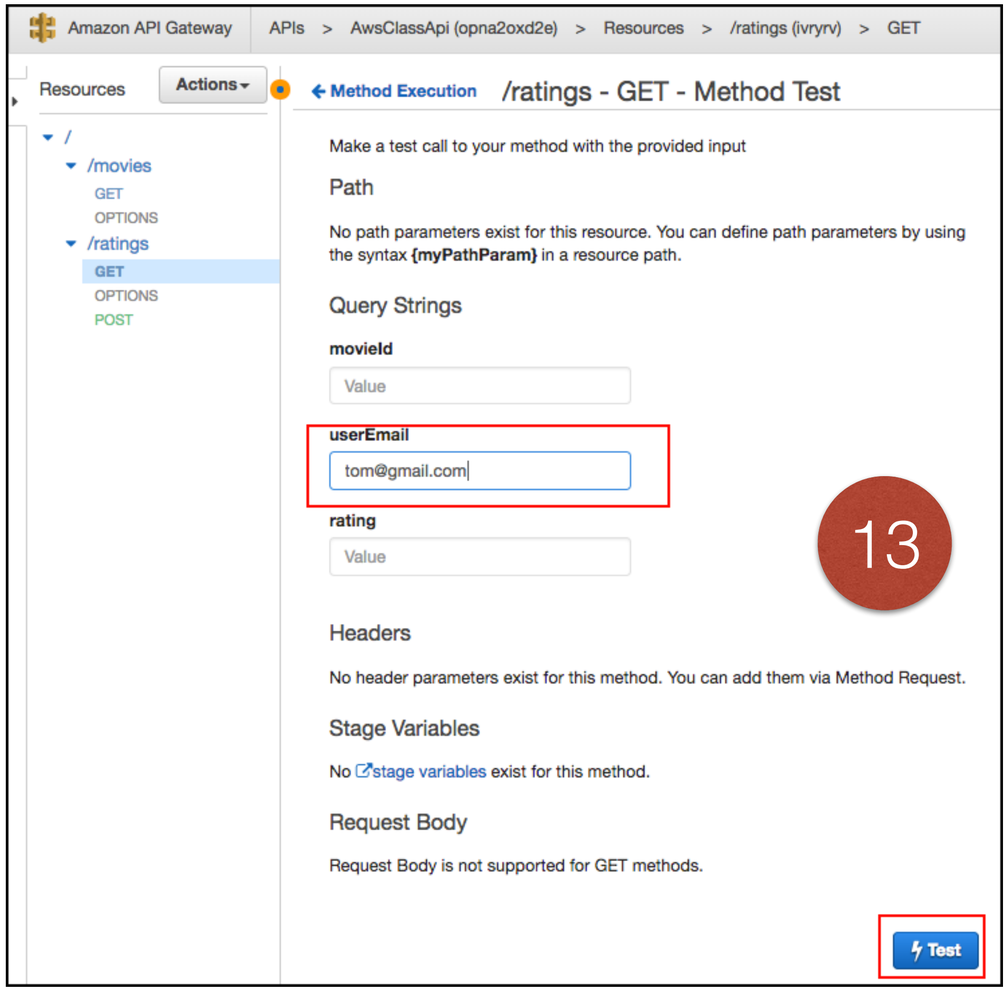




Create the GET Ratings Endpoint

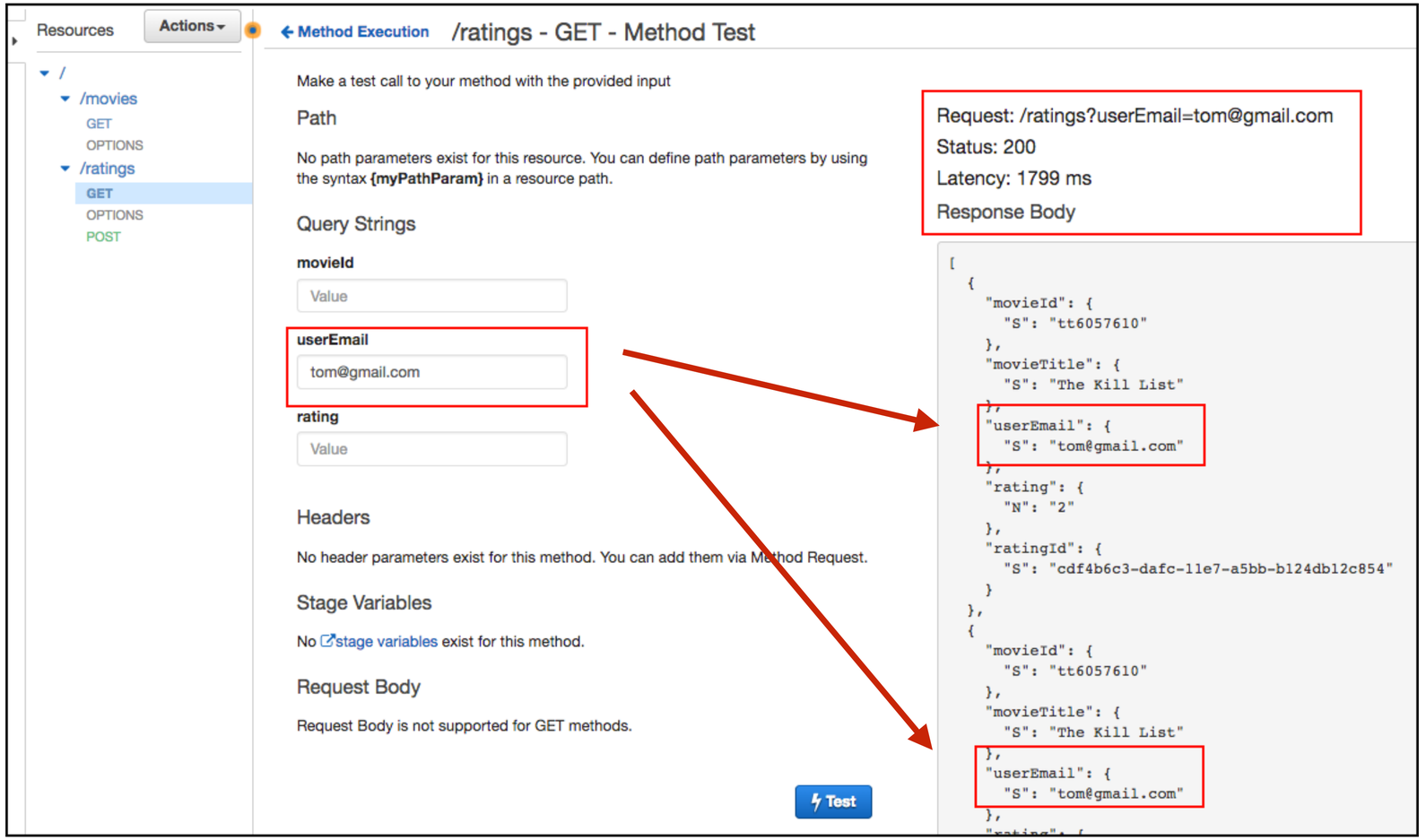
12



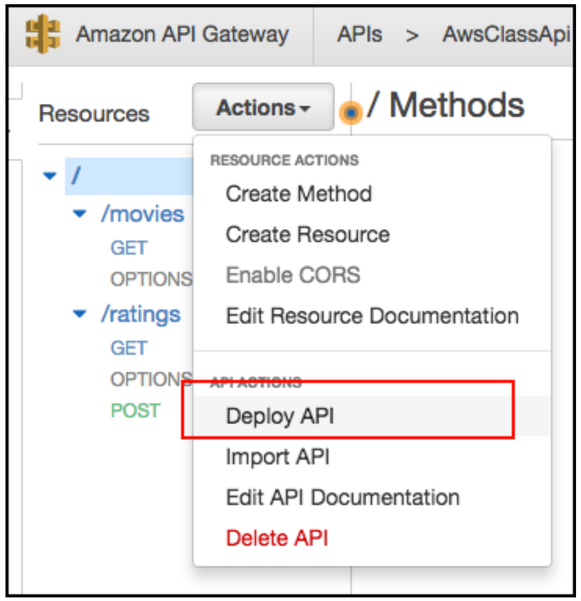


Create the GET Ratings Endpoint

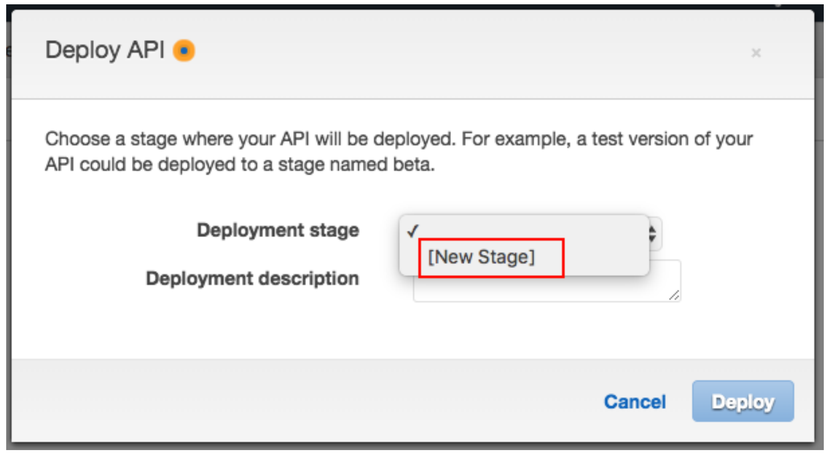
14



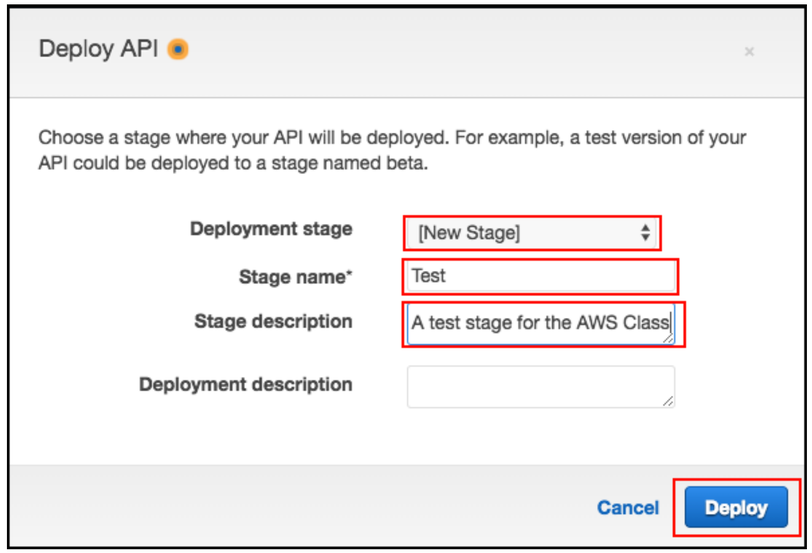
Deploy API to Test



1

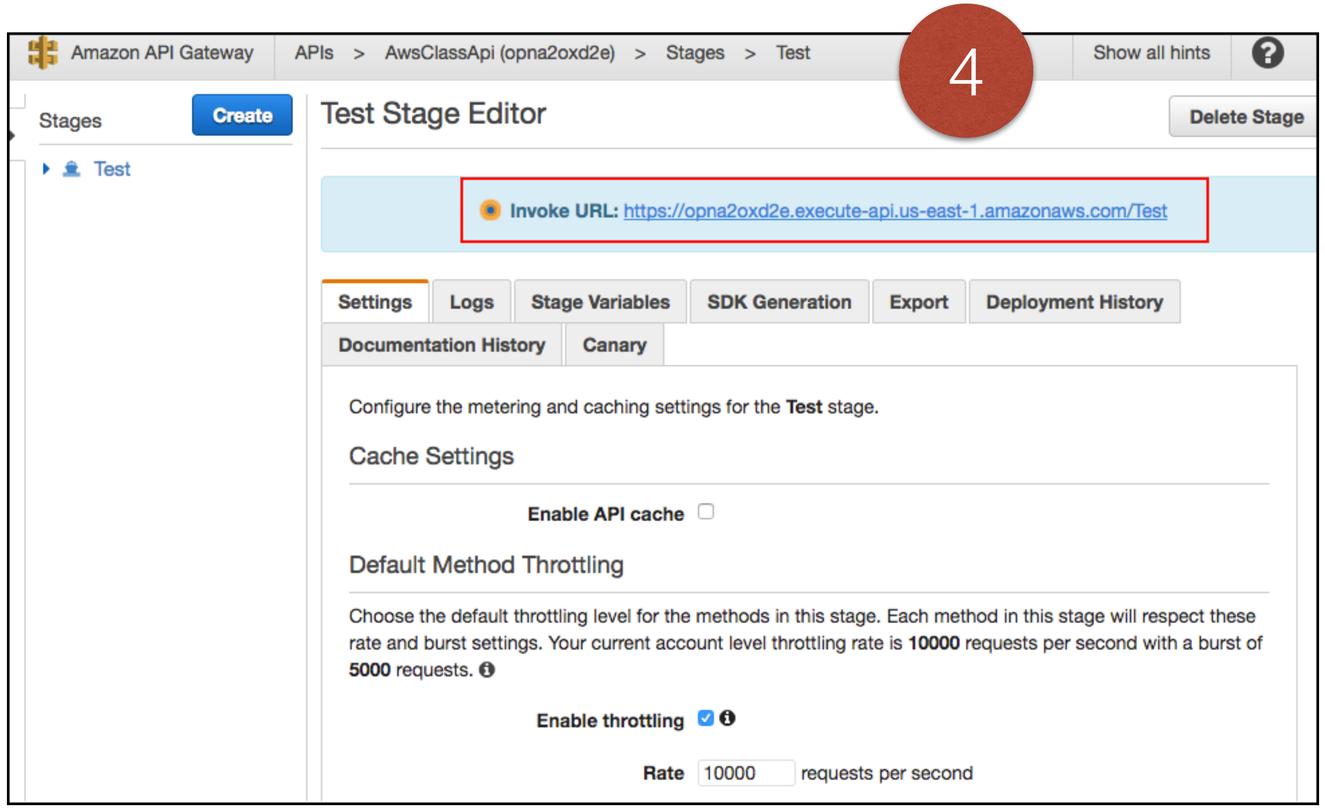


2

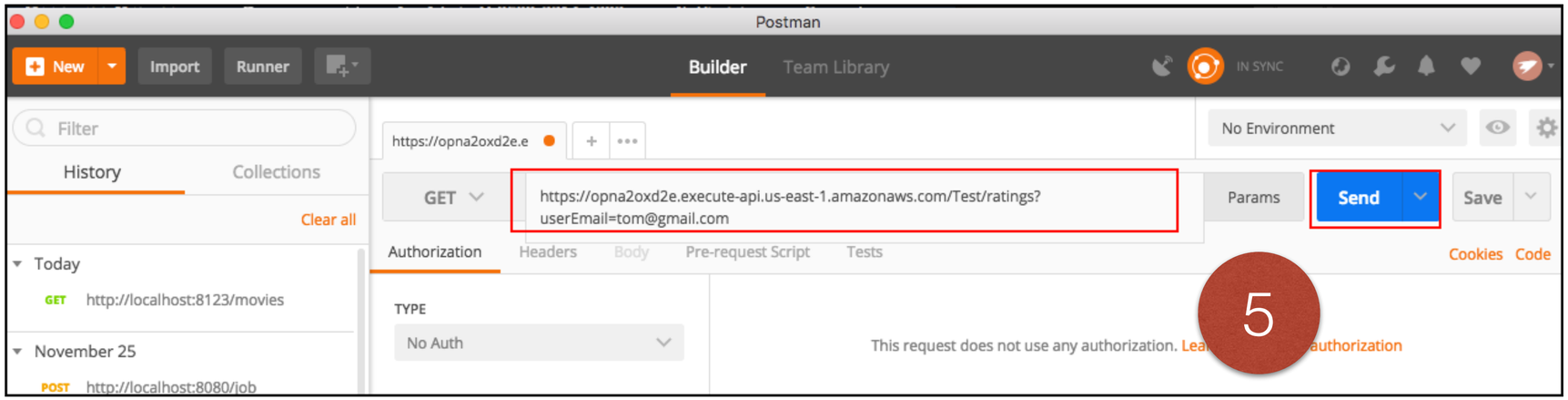


3

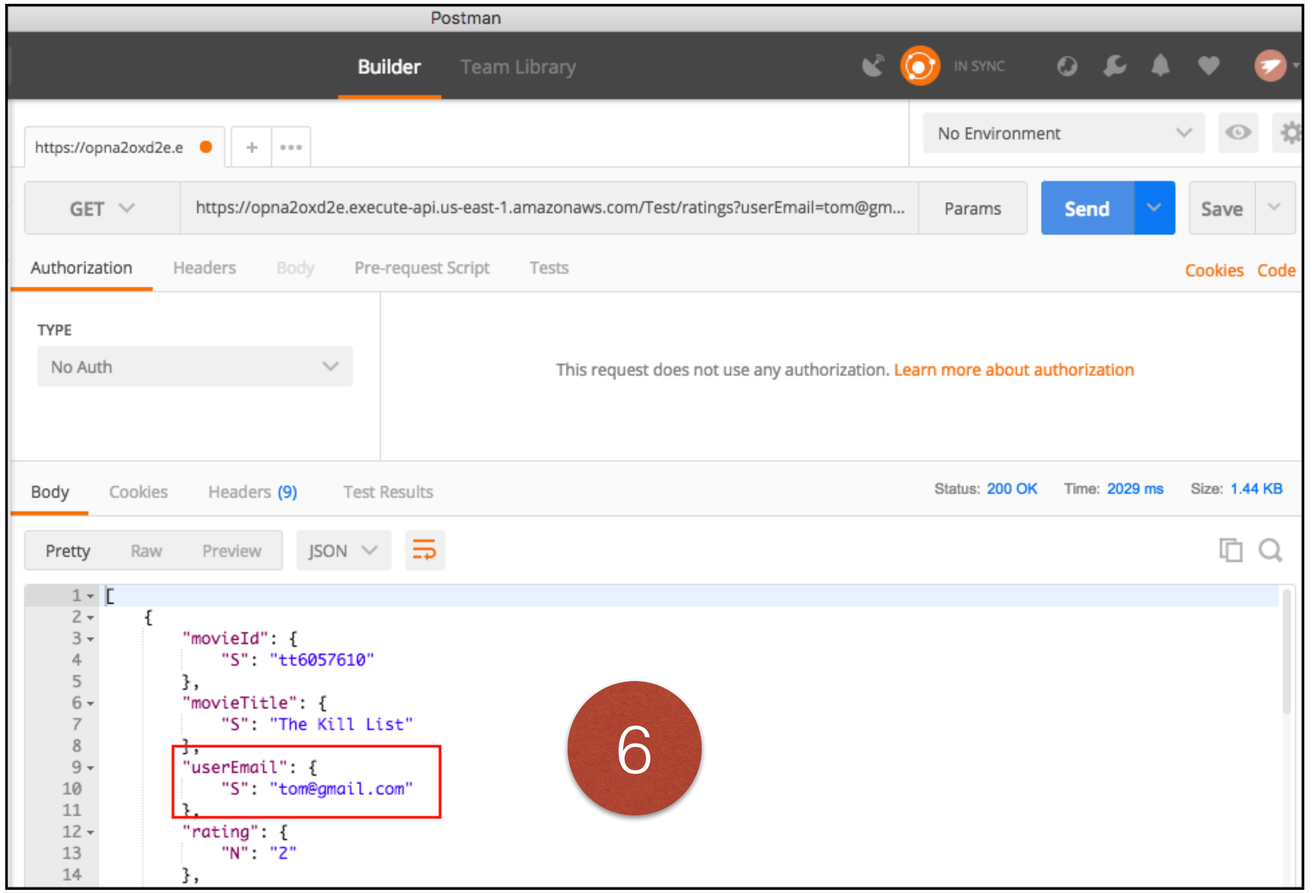
Deploy API to Test



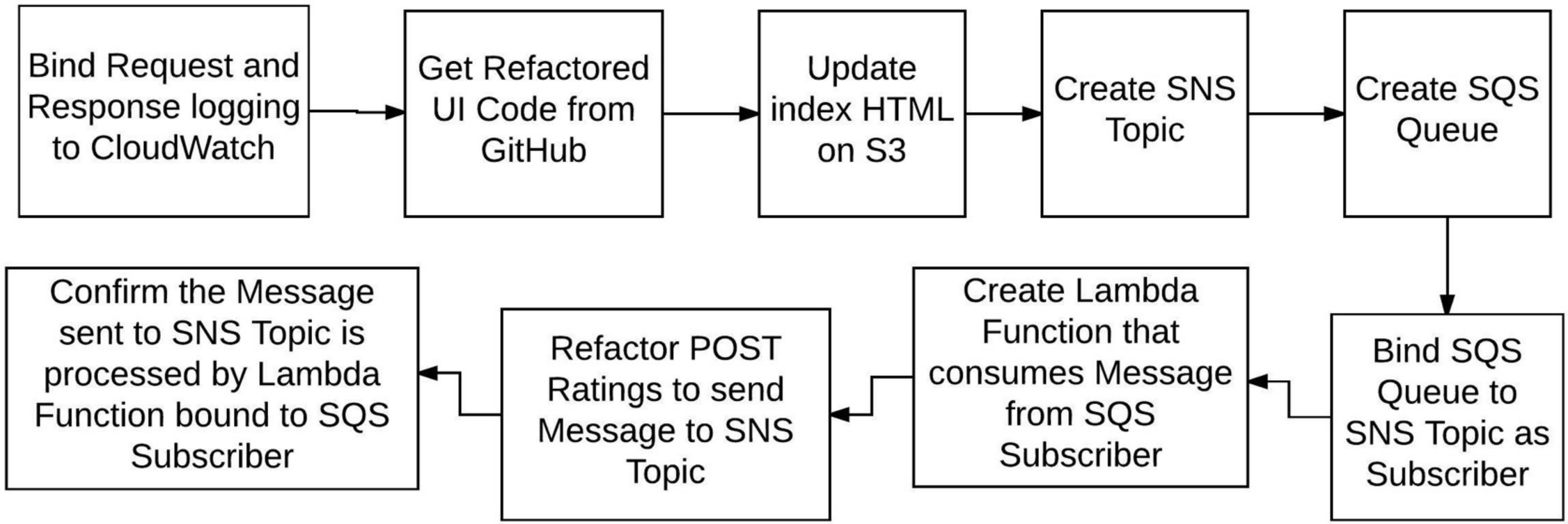
Deploy API to Test



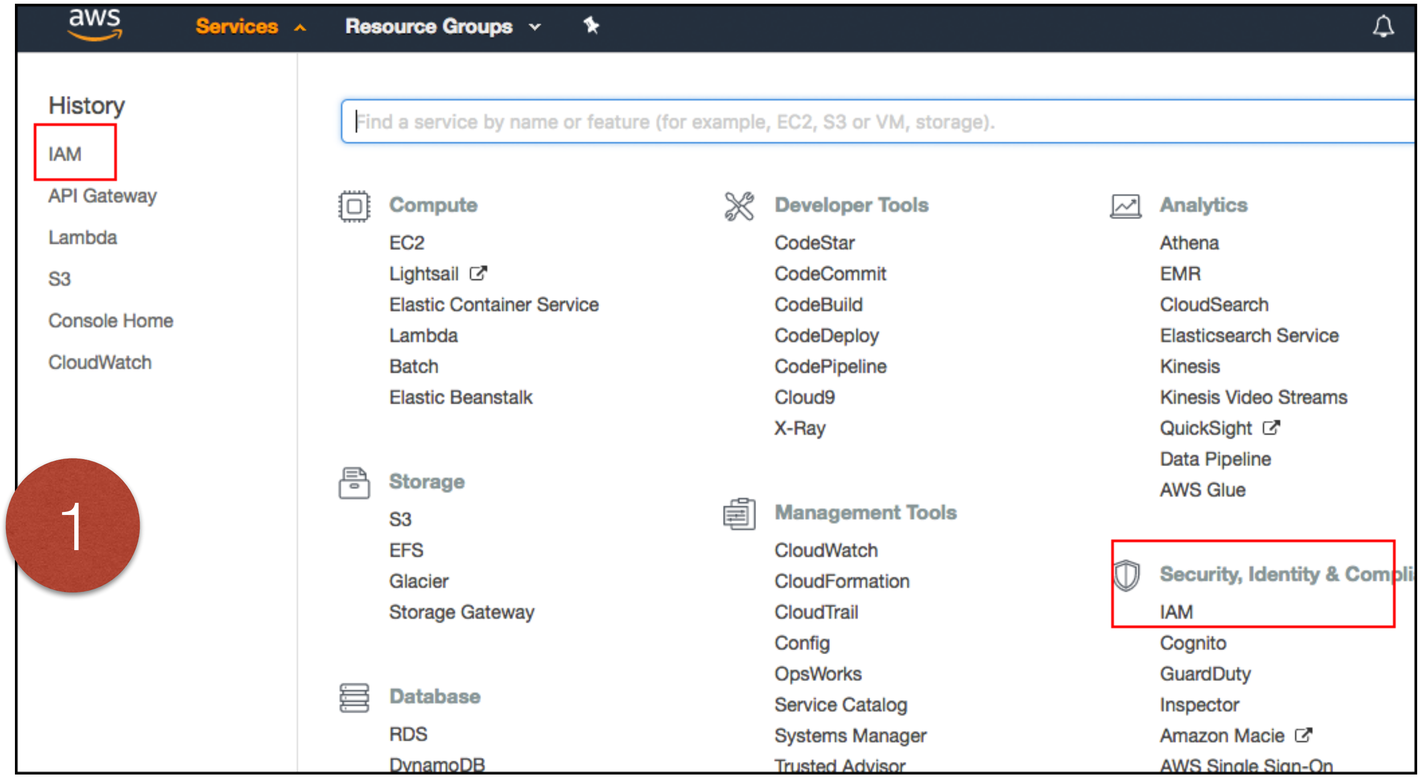
Deploy API to Test

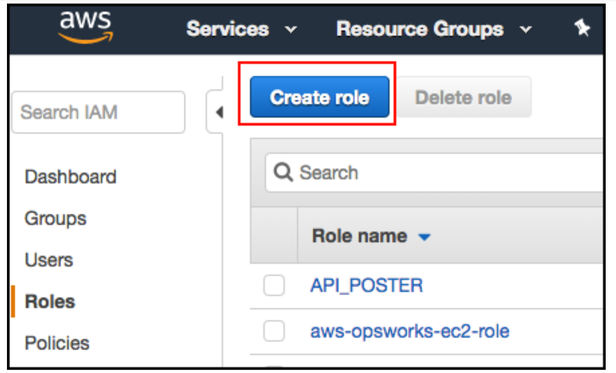


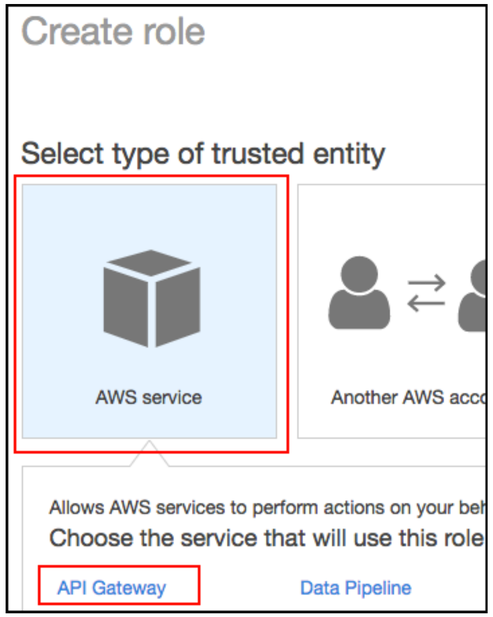
Session 6


Bind Request and Response logging to CloudWatch
Get Refactored UI Code from GitHub
Update index HTML on S3
Create SNS Topic
Create SQS Queue
Bind SQS Queue to SNS Topic as Subscriber
Create Lambda Function that consumes Message from SQS Subscriber
Refactor POST Ratings to send Message to SNS Topic
Confirm the Message sent to SNS Topic is processed by Lambda Function bound to SQS Subscriber

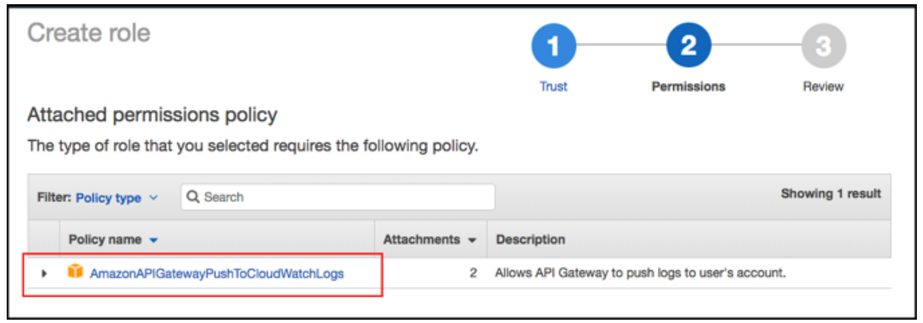
Bind Request/Response to CloudWatch



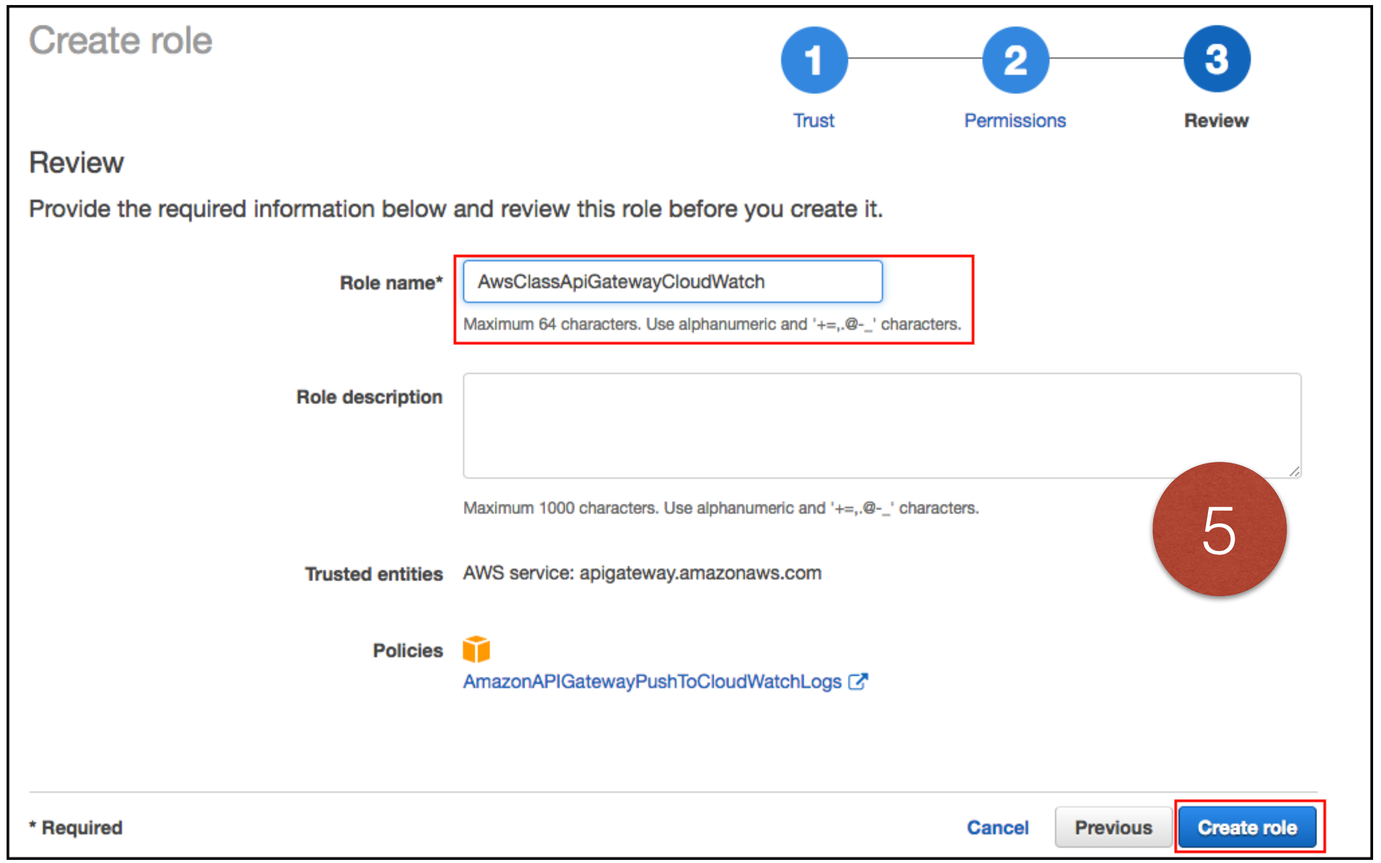




Bind Request/Response to CloudWatch

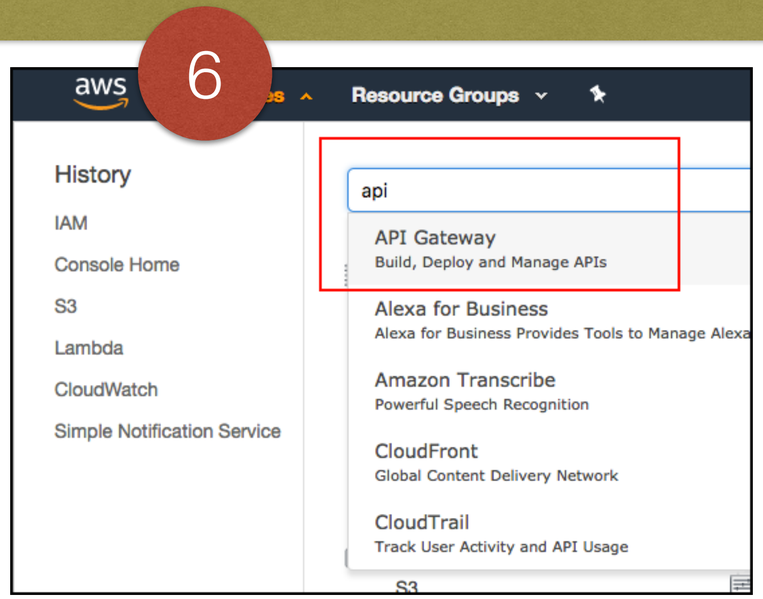


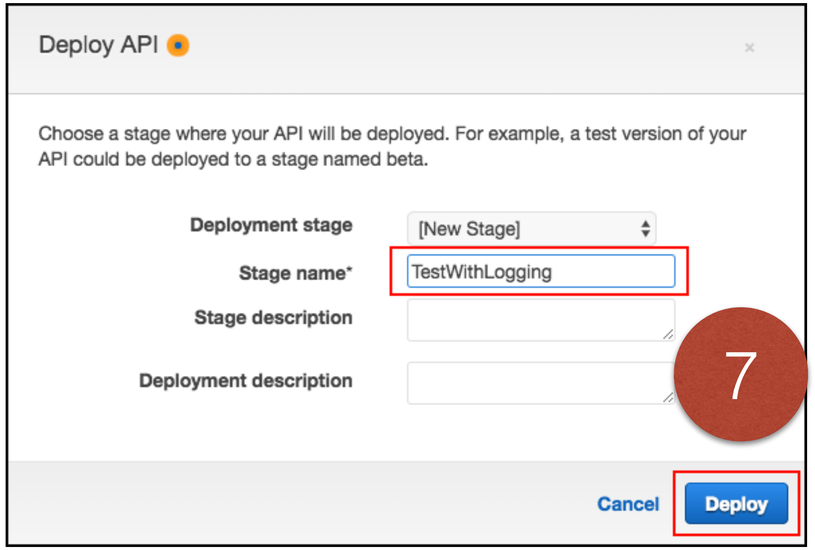
4

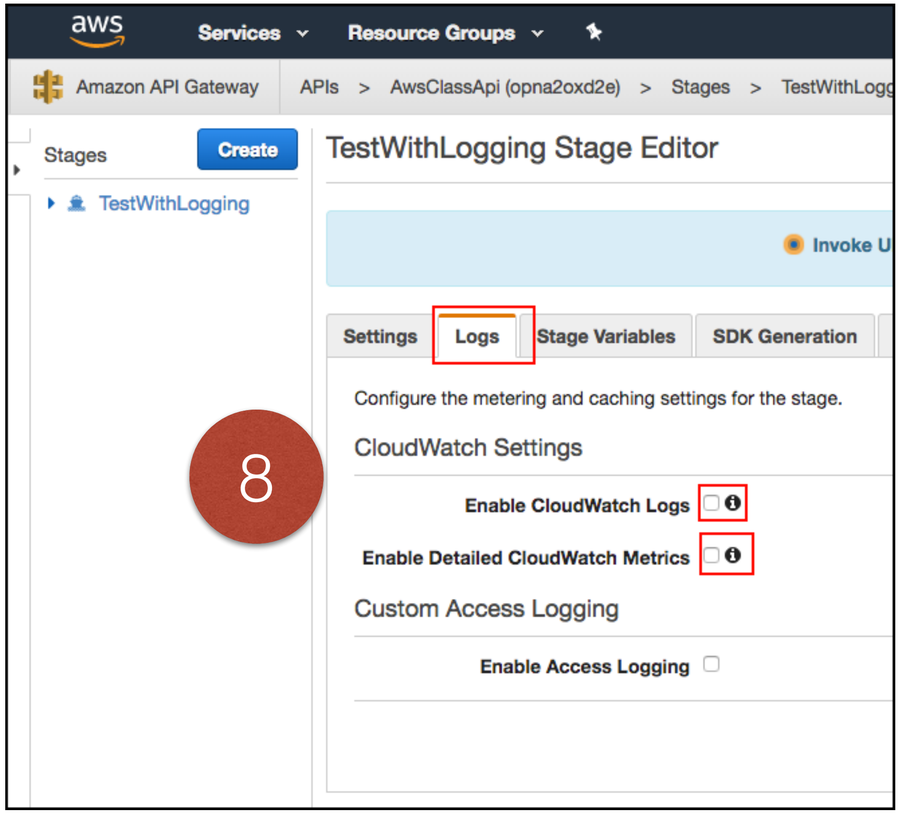


5

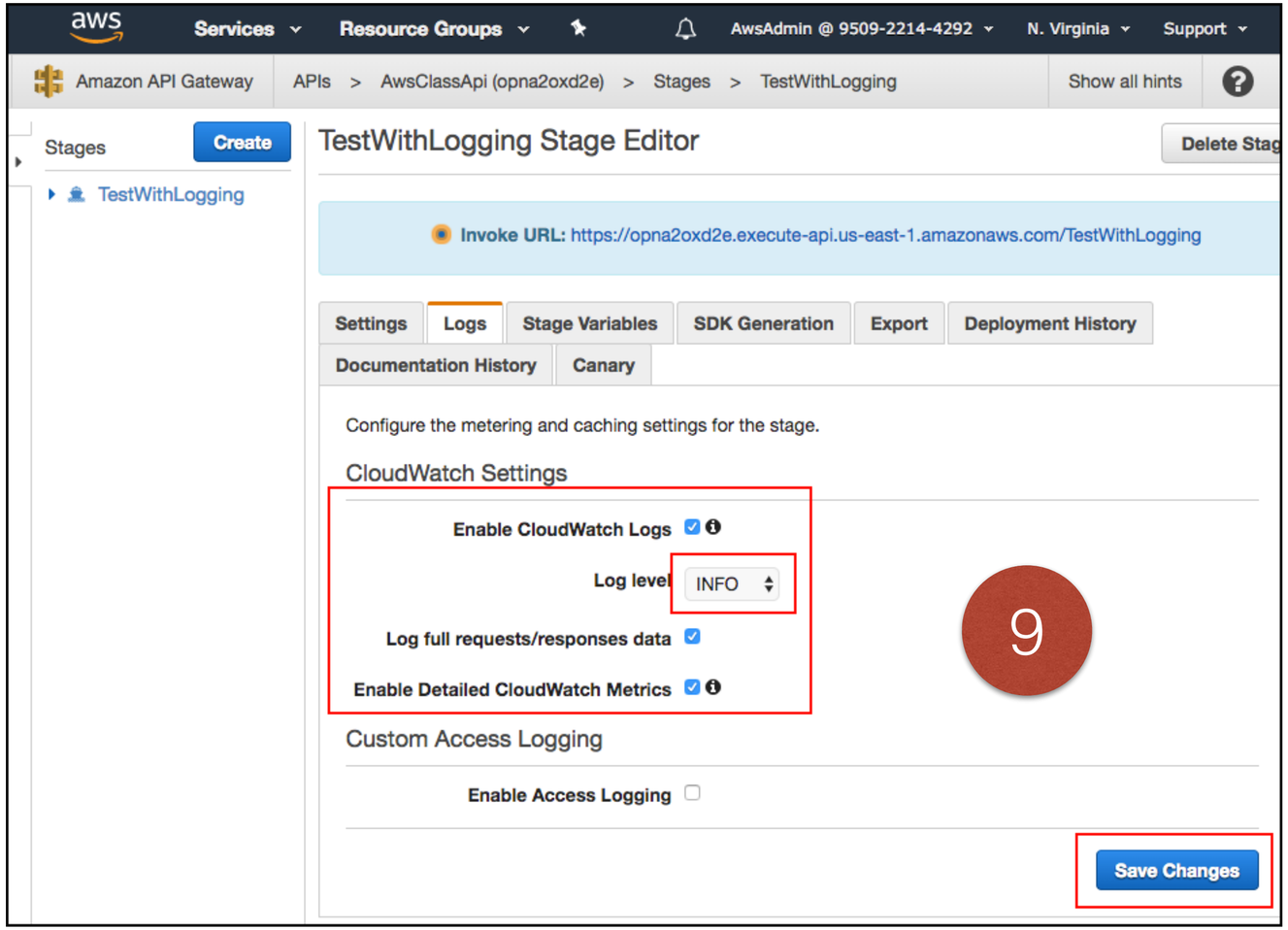
Bind Request/Response to CloudWatch



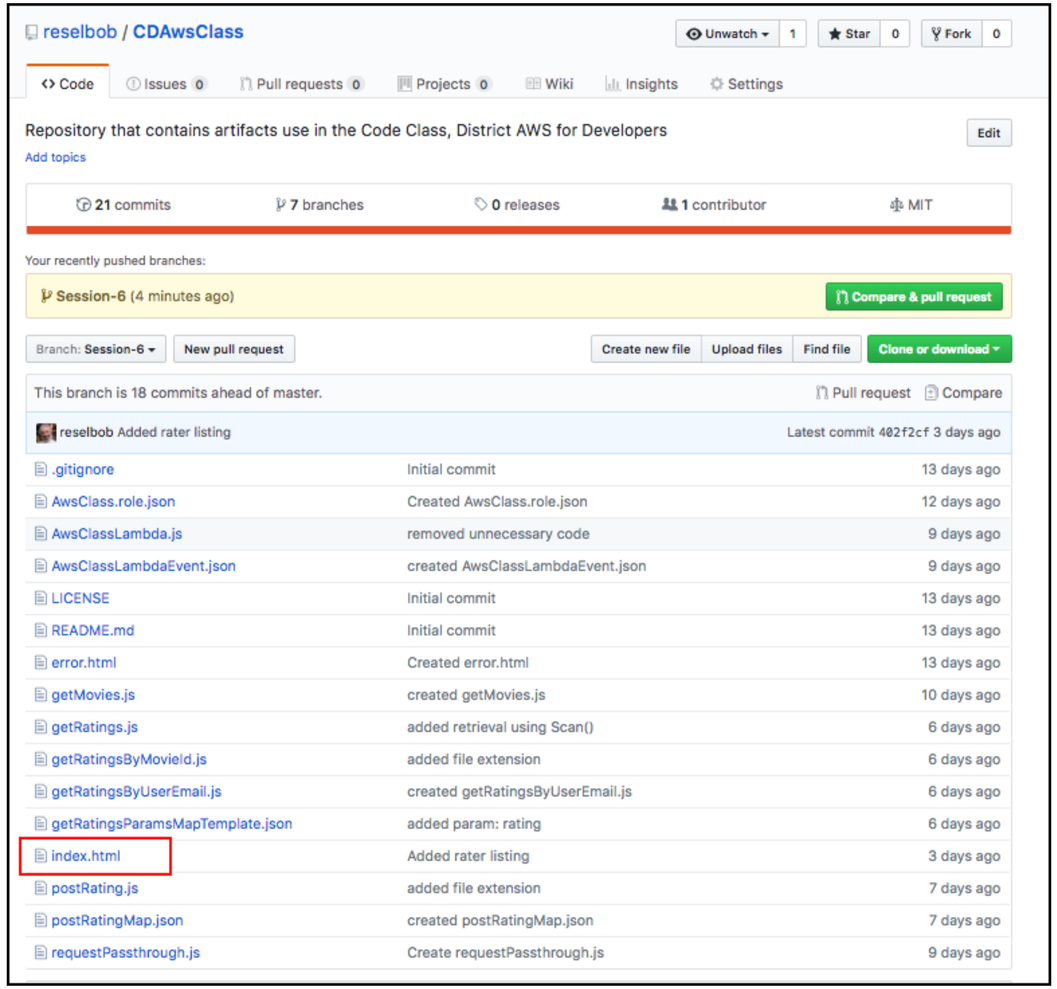




Bind Request/Response to CloudWatch



Get Refactored UI Code from GitHub



10

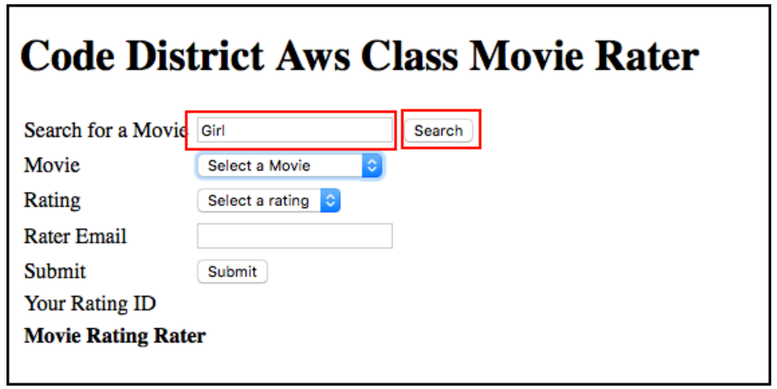
Get Refactored UI Code from GitHub

```
</table>
<script>

    var API_URL = 'https://opna2oxd2e.execute-api.us-east-1.amazonaws.com/Test';


    function populateRatingTable(data){

        //alert(JSON.stringify(data));
        var tbl = document.getElementById("myRatings");
        data.forEach(function(row){
            var str = '<td>' + row.movieTitle.S + '</td>';
            str += '<td>' + row.rating.N + '</td>'
            str += '<td>' + row.userEmail.S + '</td>'
            $("#myRatings").last().append('<tr>' + str + '</tr>');
        });
    }
```

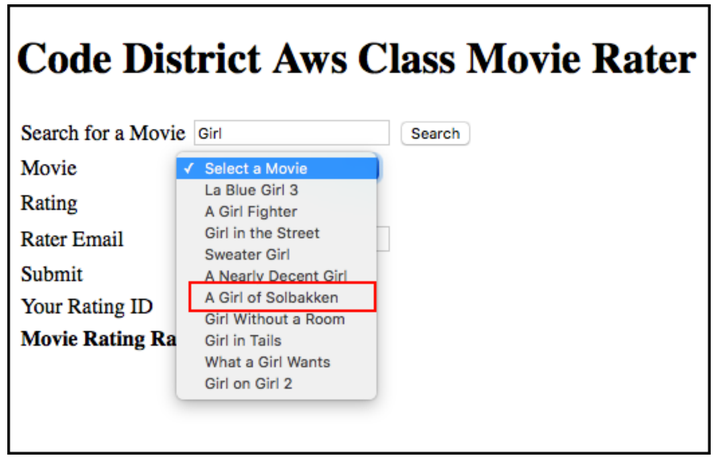
11

Change to **TestWithLogging**

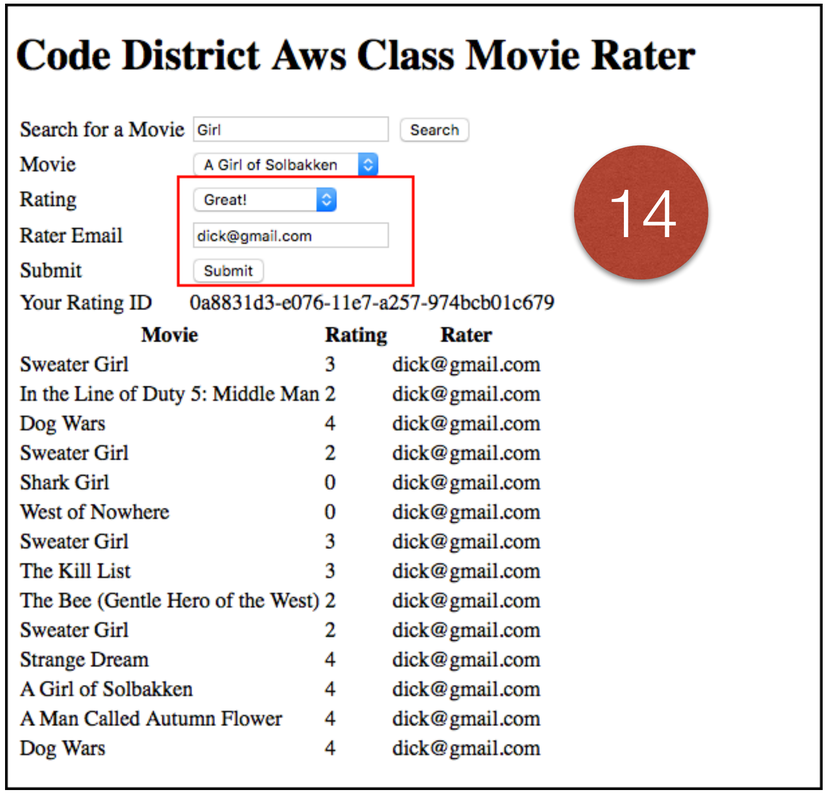
Get Refactored UI Code from GitHub



12

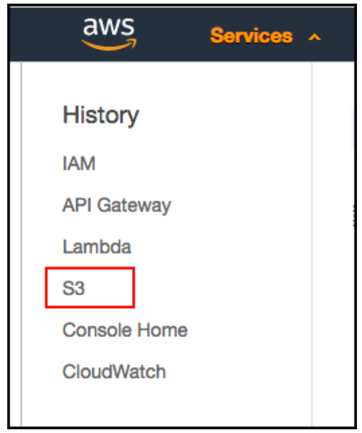


13

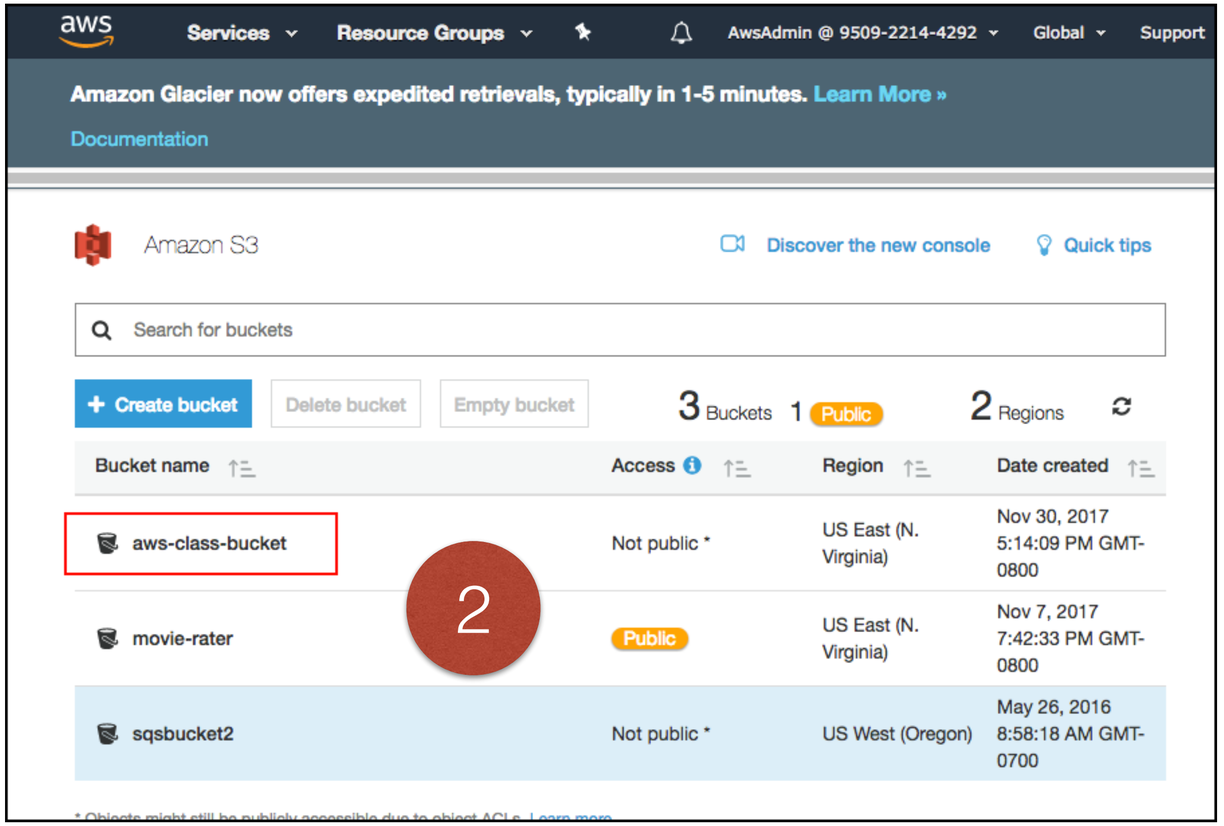


14

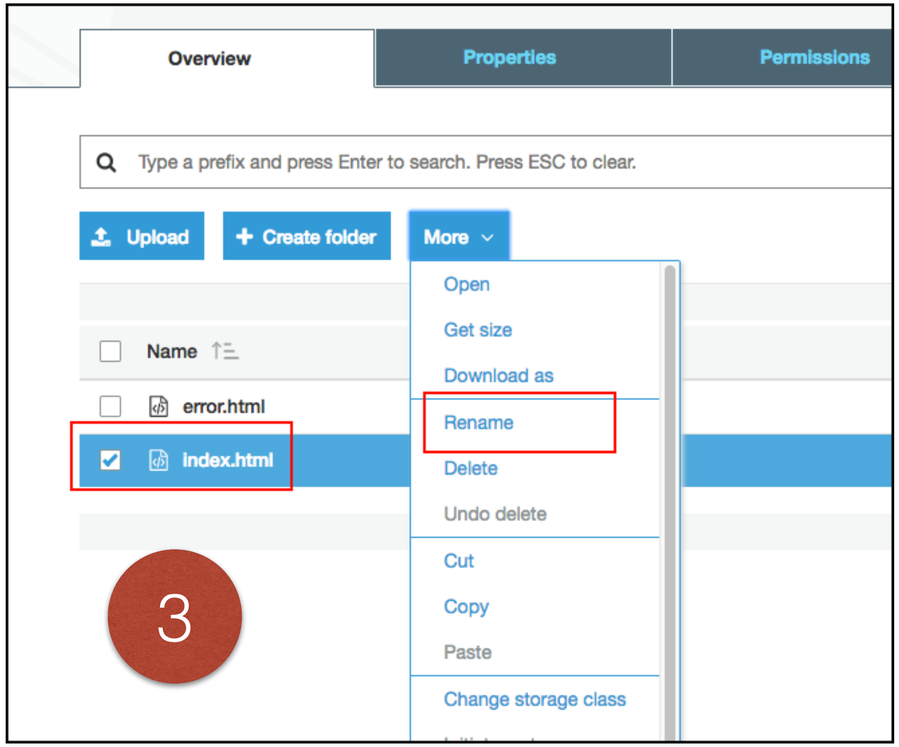
Update Index HTML on S3



1

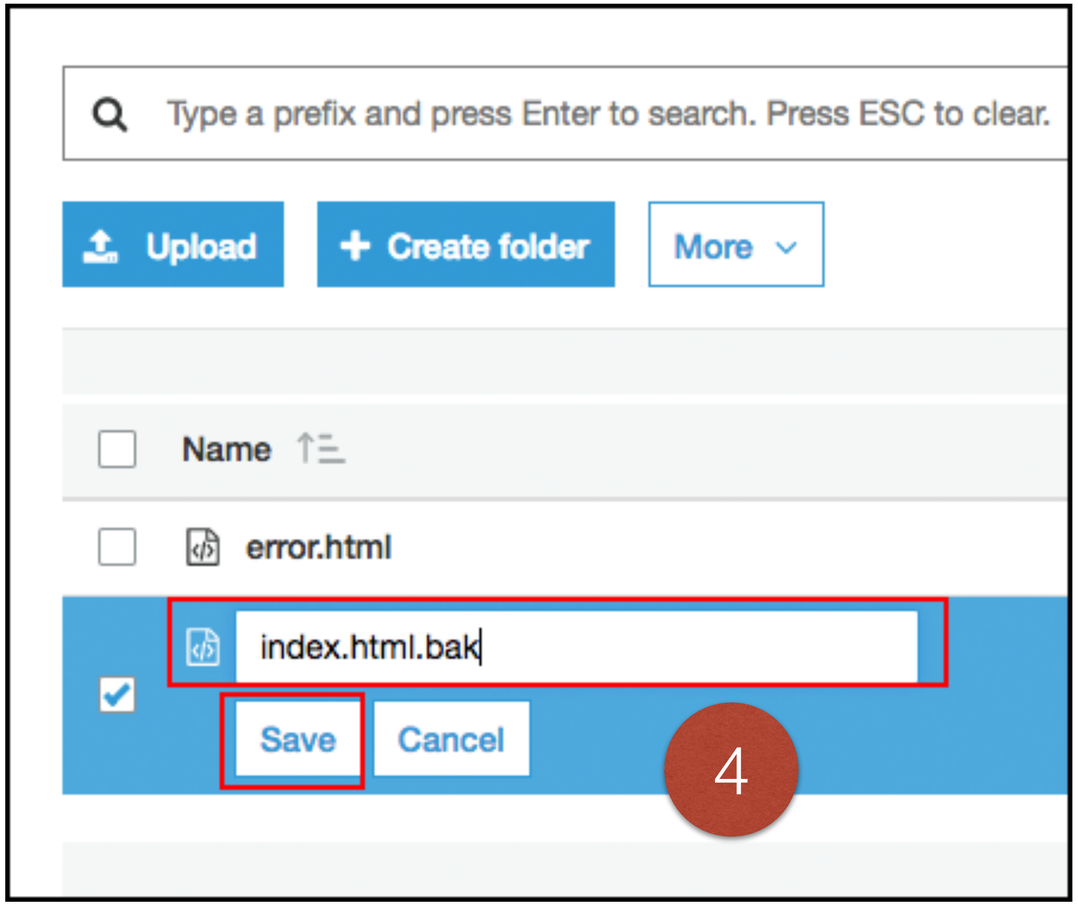


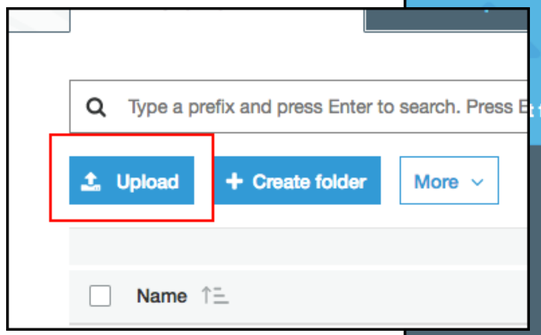
2

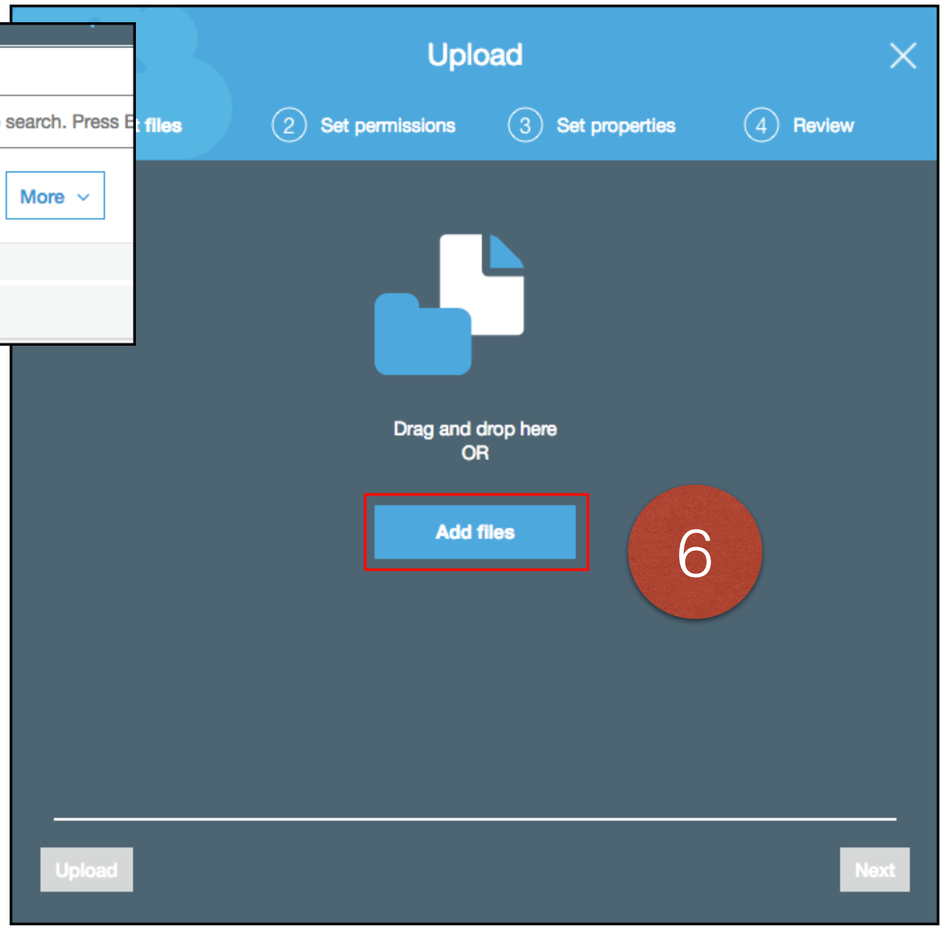


3

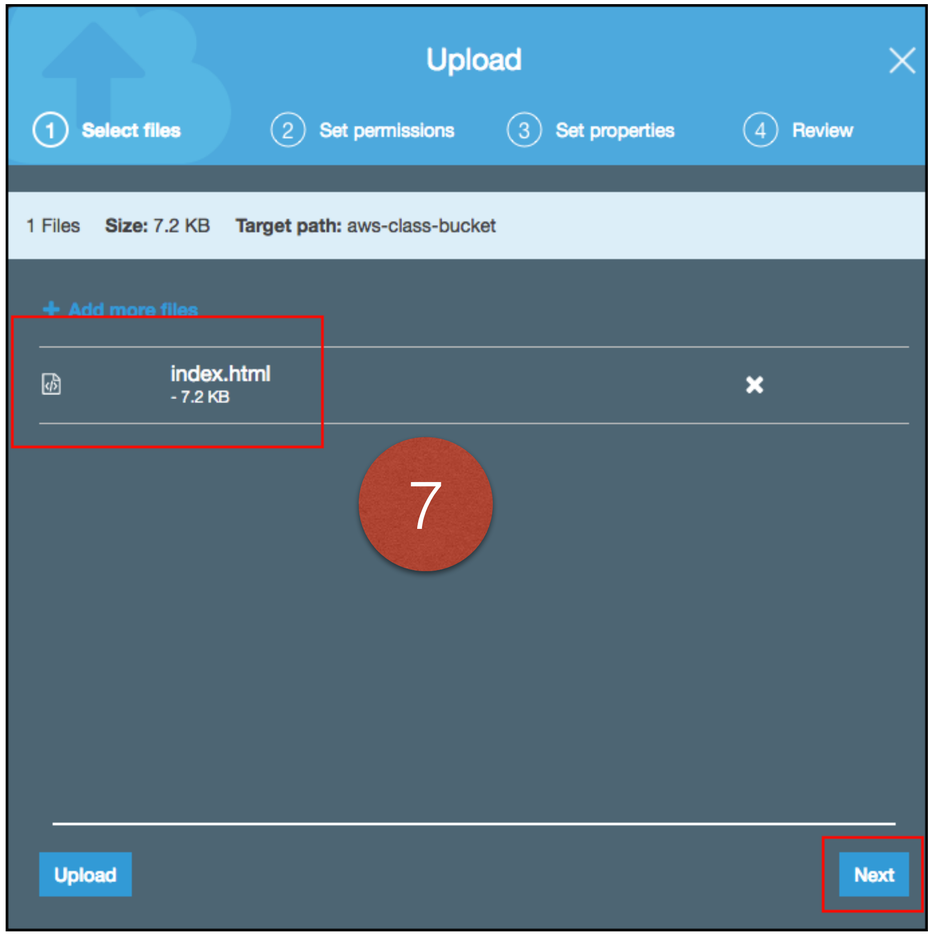
Update Index HTML on S3

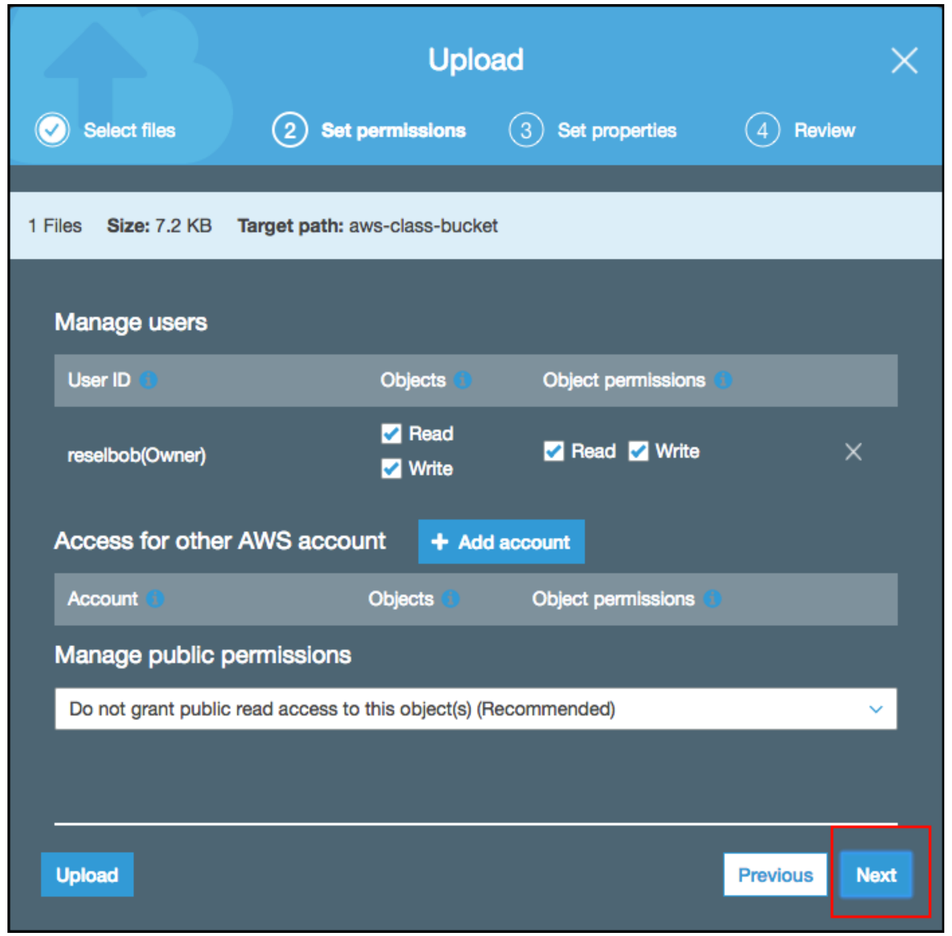




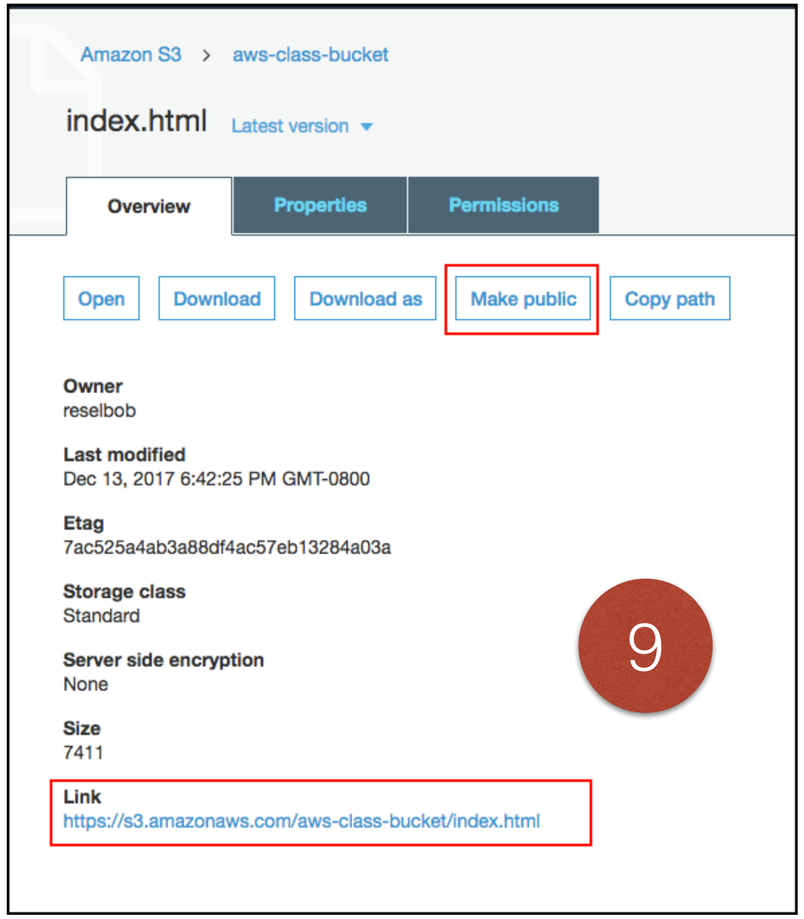


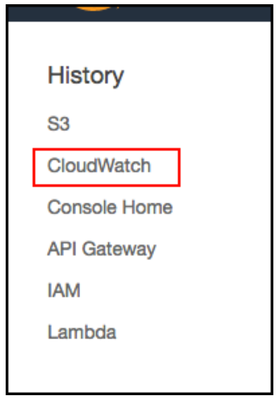
Update Index HTML on S3

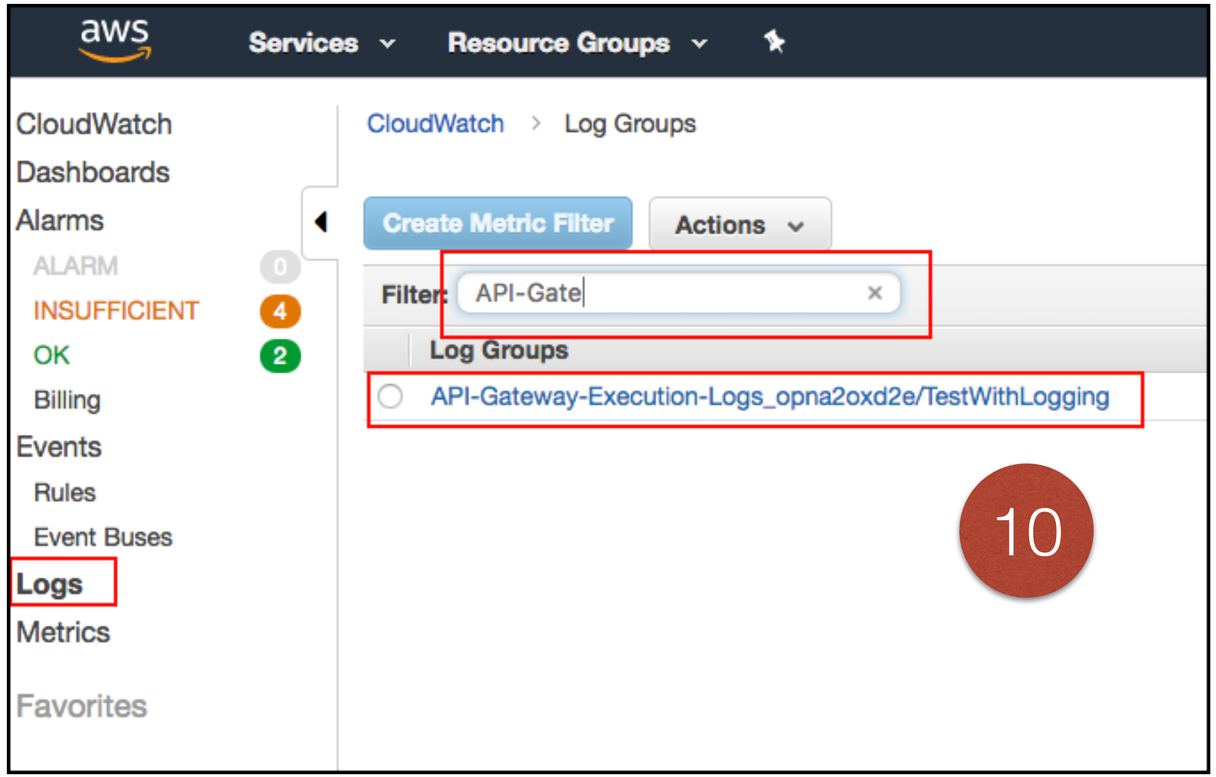




Update Index HTML on S3

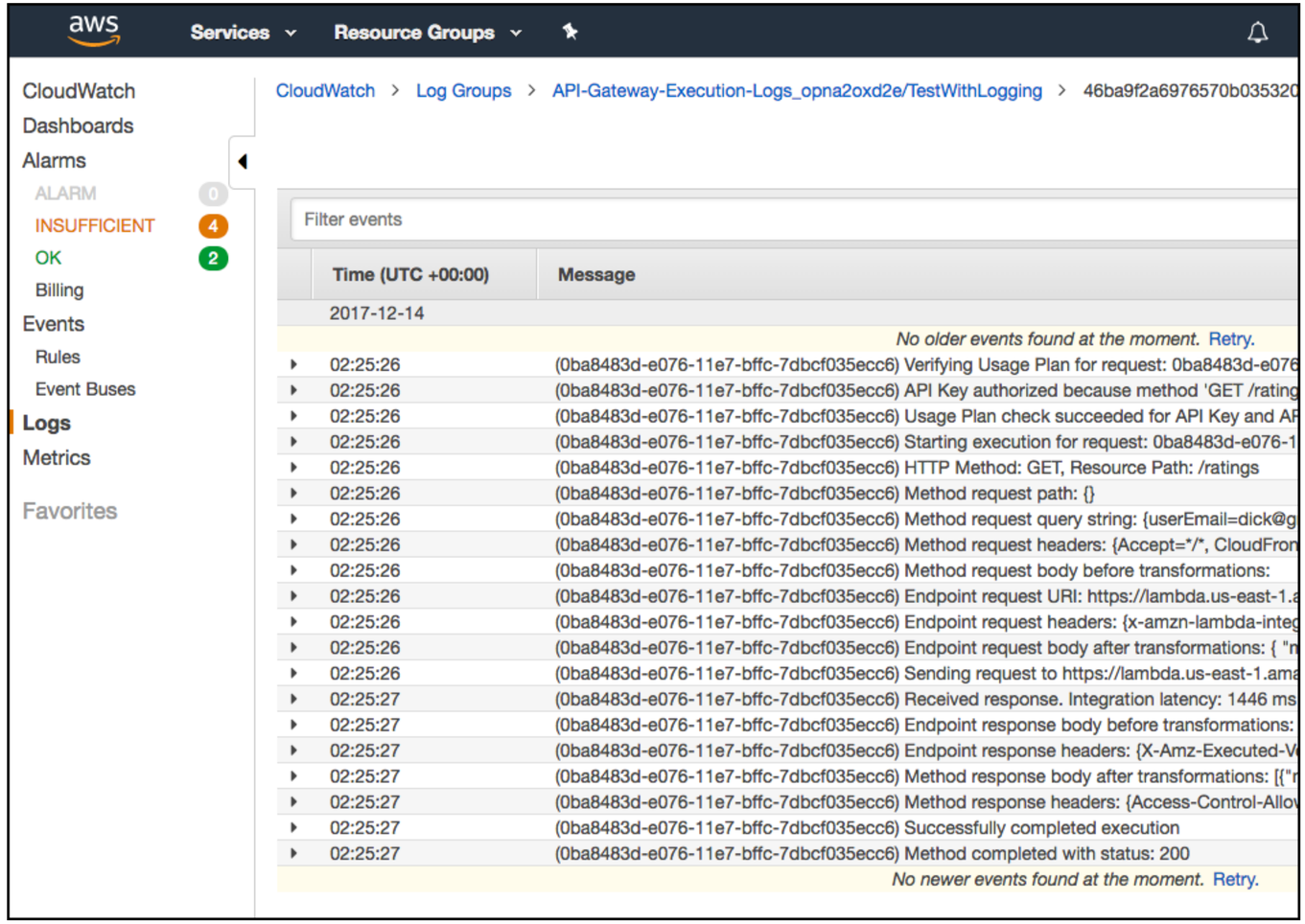




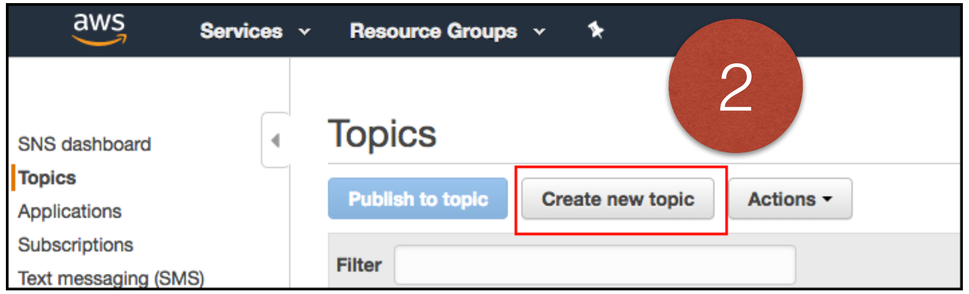


Update Index HTML on S3

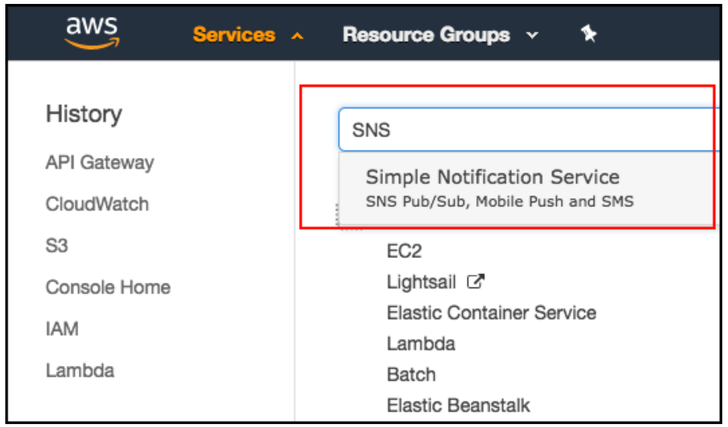
11

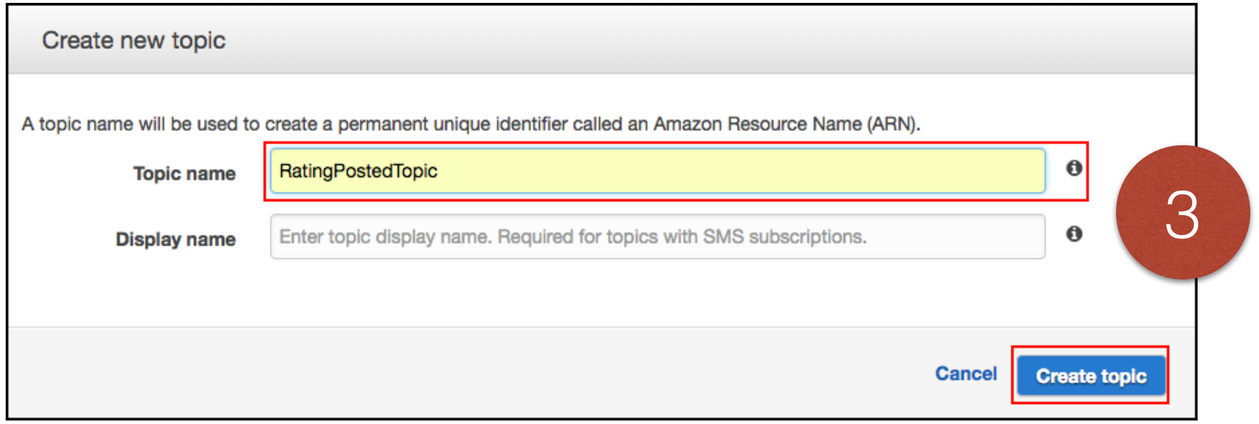


Bind SQS Queue to SNS Topic as Subscriber

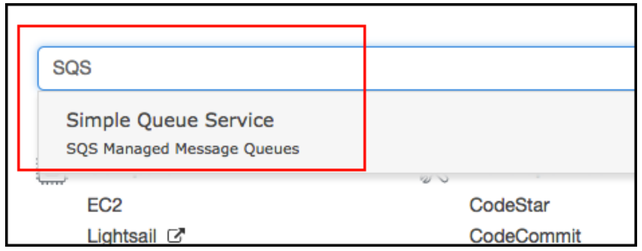


1

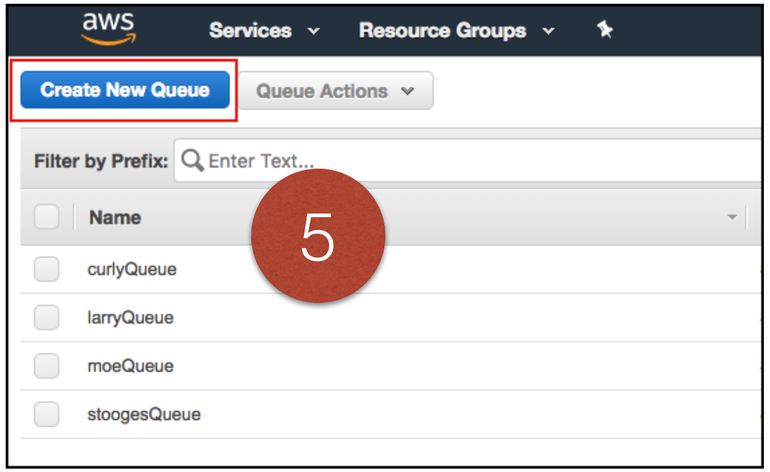




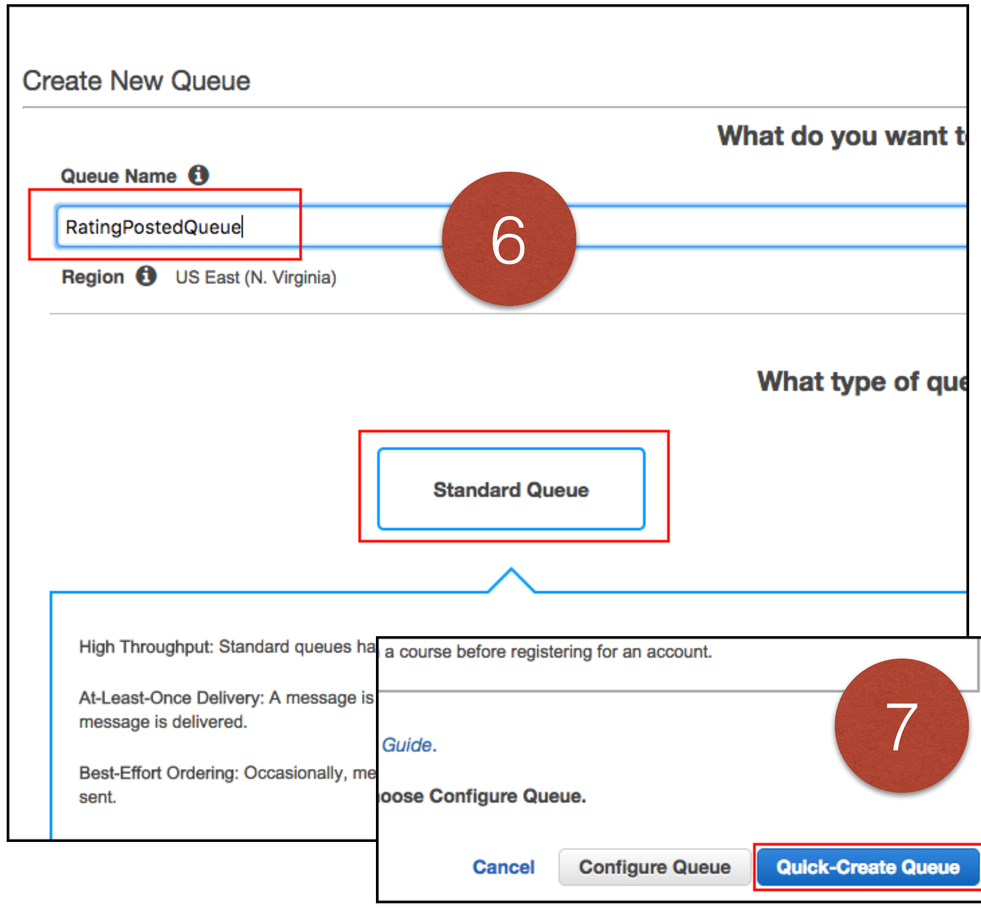
Bind SQS Queue to SNS Topic as Subscriber



4



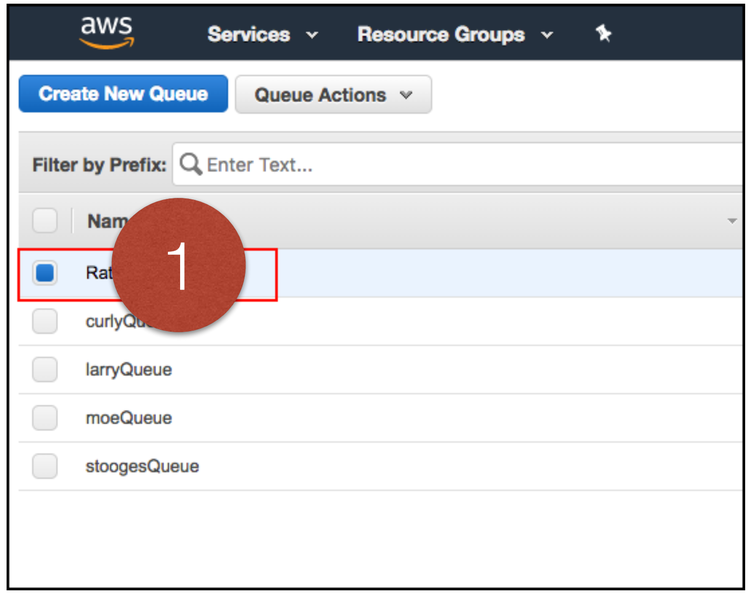
5

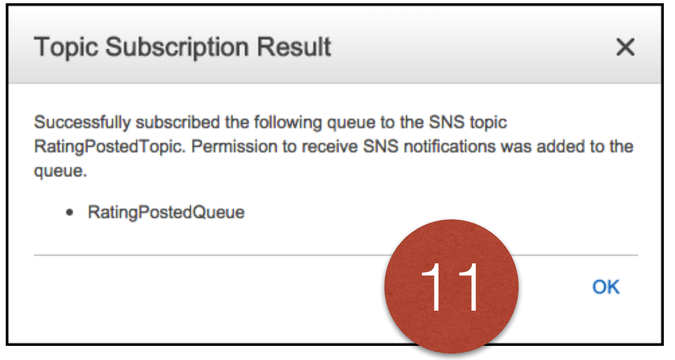


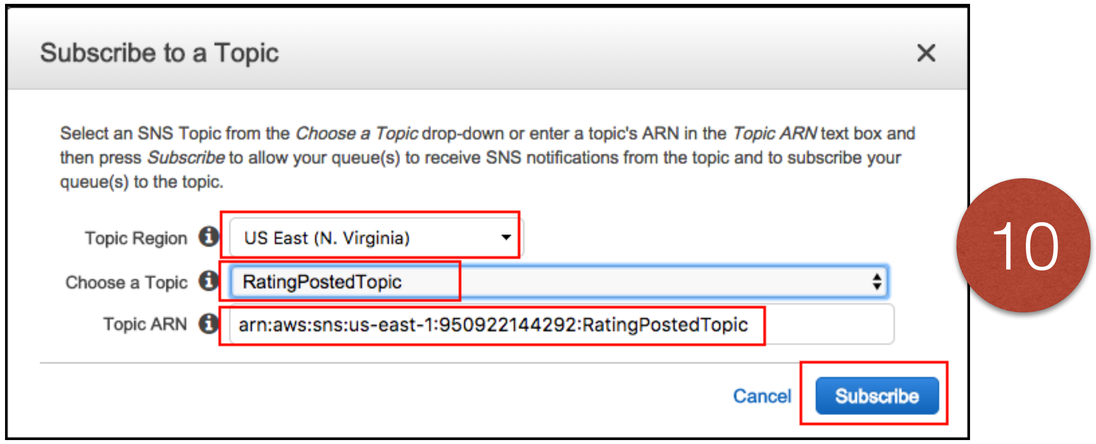
6

7

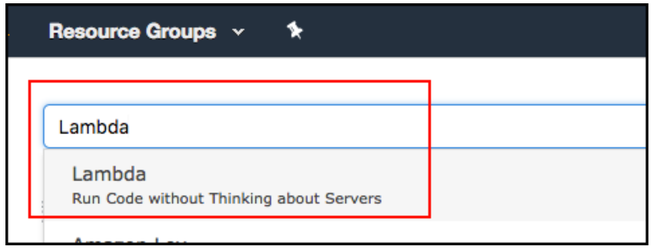
Bind SQS Queue to SNS Topic as Subscriber



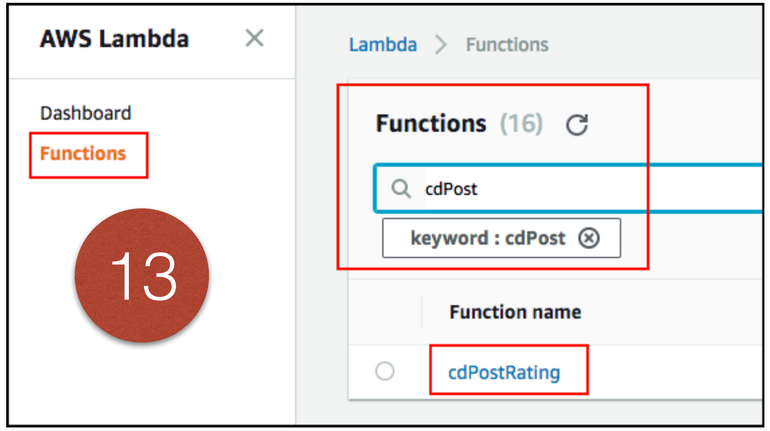




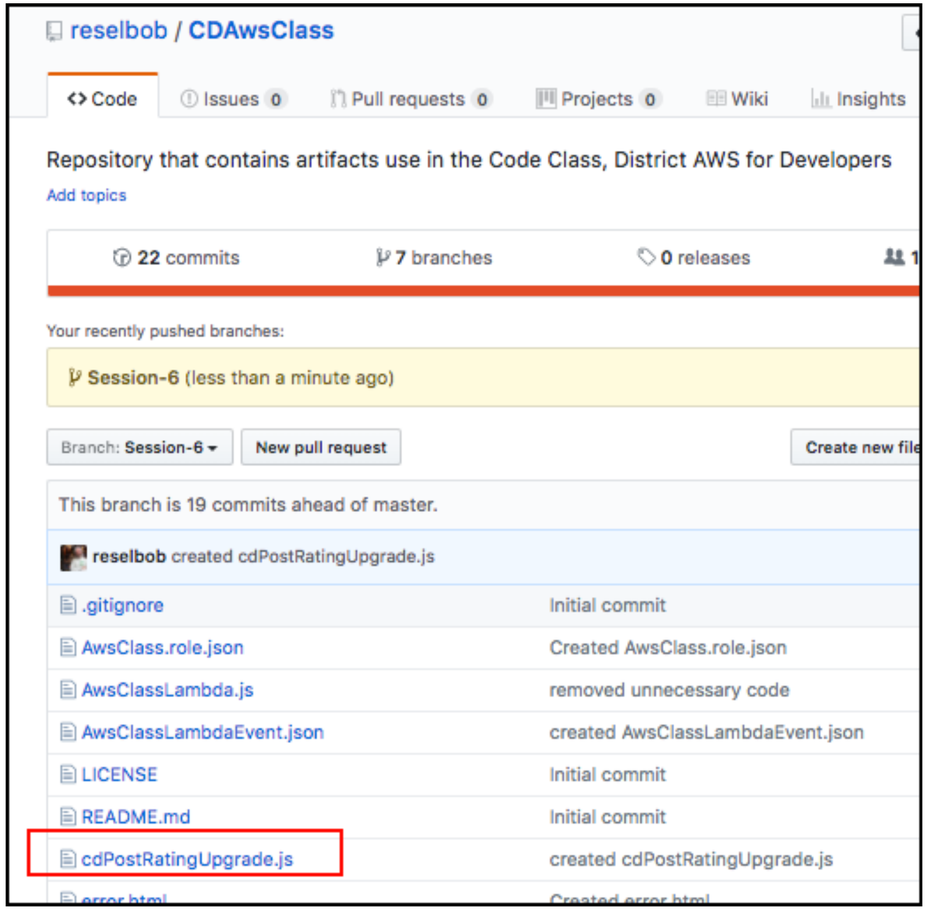
Bind SQS Queue to SNS Topic as Subscriber



12



13



14

Bind SQS Queue to SNS Topic as Subscriber

```
24  /*
25  sends the rating onto the SNS Topic
26  */
27  function sendRatingToTopic(rating, callback){
28      var sns = new AWS.SNS();
29       console.log({message: 'Send sending rating to topic: arn:aws:sns:us-east-1:950922144292:RatingPostedTopic.',rating });
30      sns.publish({
31        TopicArn: "arn:aws:sns:us-east-1:950922144292:RatingPostedTopic",
32        Subject: "From cdPostRating",
33        Message: JSON.stringify(rating)
34        }, callback);
35  }
```

15

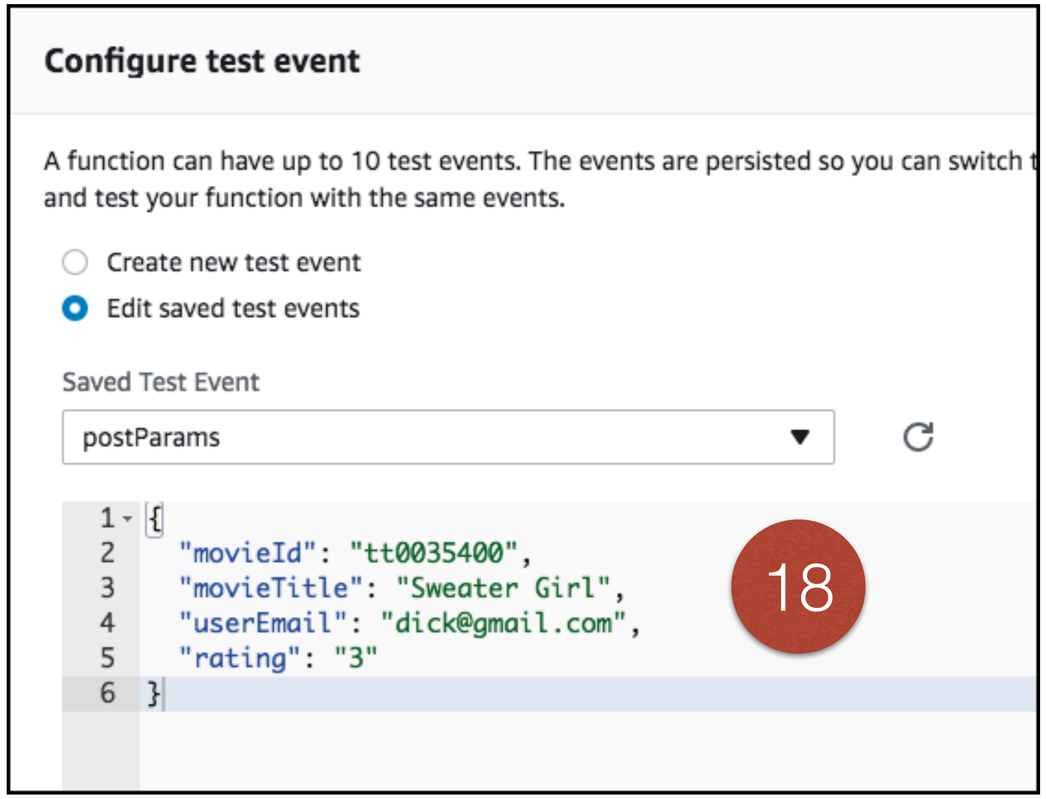
```
42      const rating = {
43              "movieId": {"S": event.movieId },
44              "movieTitle": {"S":event. movieTitle },
45              "userEmail": {"S": event.userEmail},
46              "rating": {"N": event.rating},
47              "ratingId": {"S": ratingId},
48          };
49
```

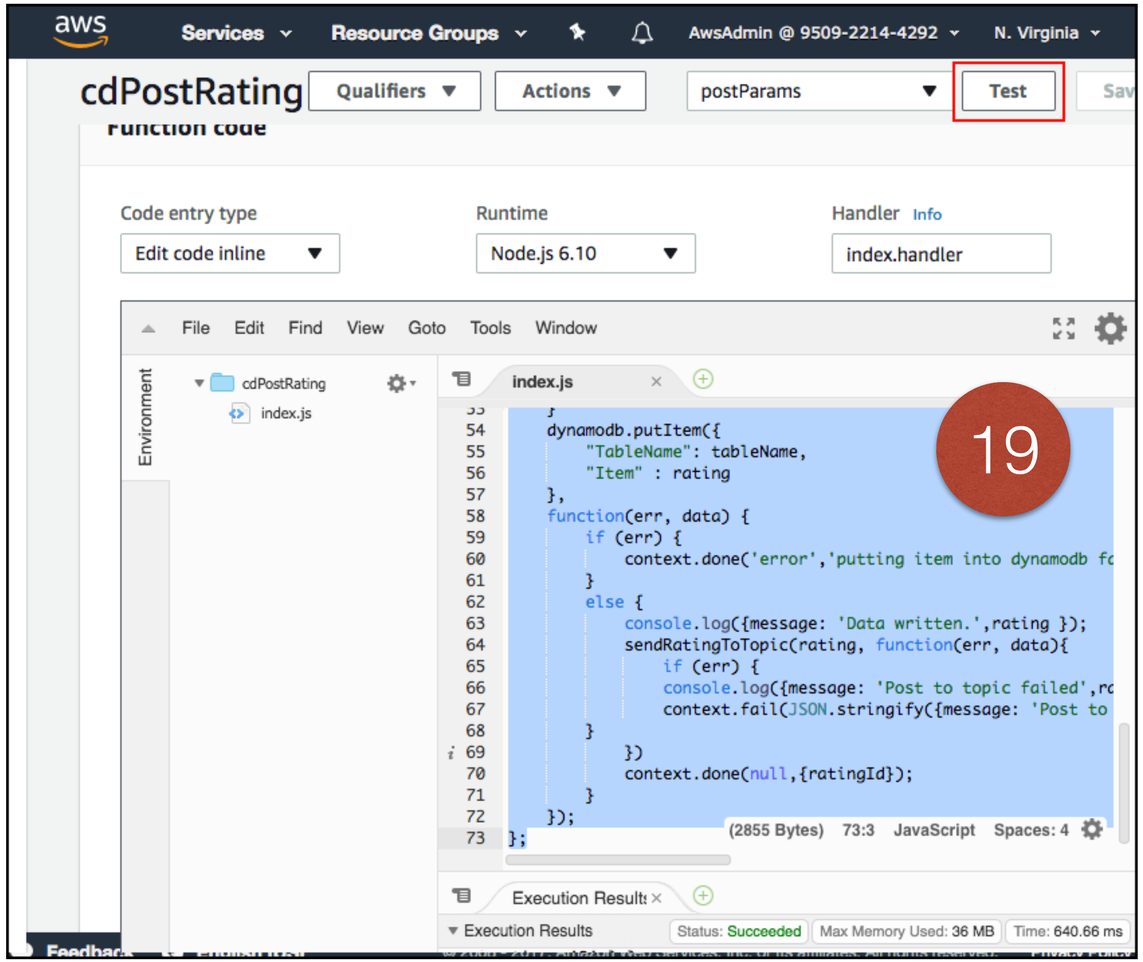
16

```
54      dynamodb.putItem({
55          "TableName": tableName,
56          "Item" : rating
57      },
58      function(err, data) {
59          if (err) {
60              context.done('error','putting item into dynamodb failed: '+ err);
61          }
62          else {
63              console.log({message: 'Data written.',rating });
64              sendRatingToTopic(rating, function(err, data){
65                  if (err) {
66                  console.log({message: 'Post to topic failed',rating, err });
67                  context.fail(JSON.stringify({message: 'Post to topic failed',rating, err}));
68              }
69                  })
70                  context.done(null,{ratingId});
71              }
72      });
```

17

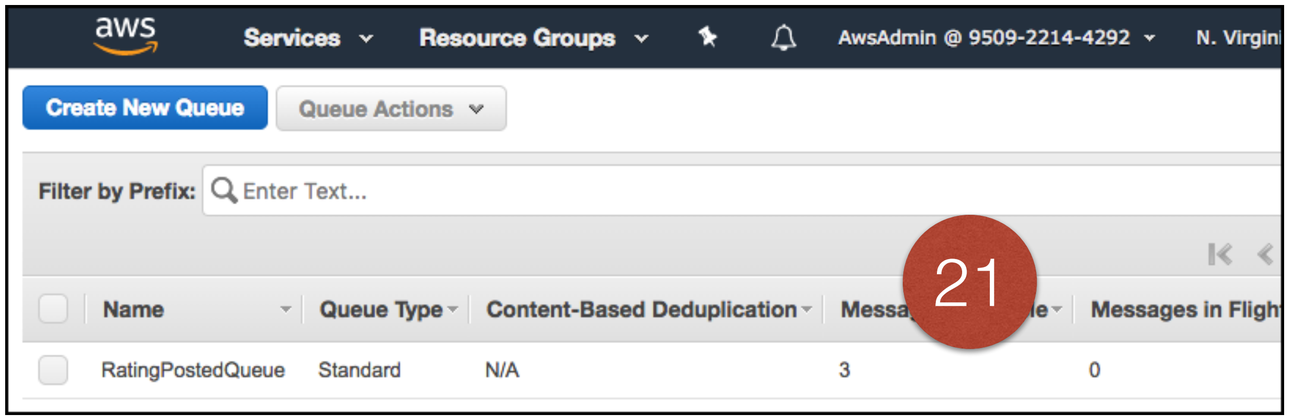
Bind SQS Queue to SNS Topic as Subscriber



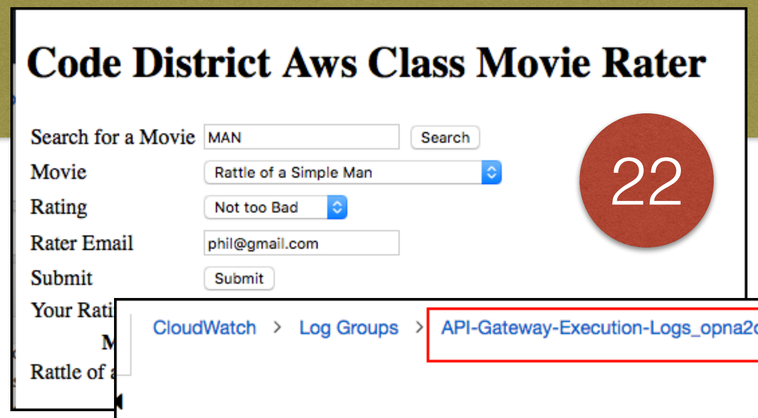


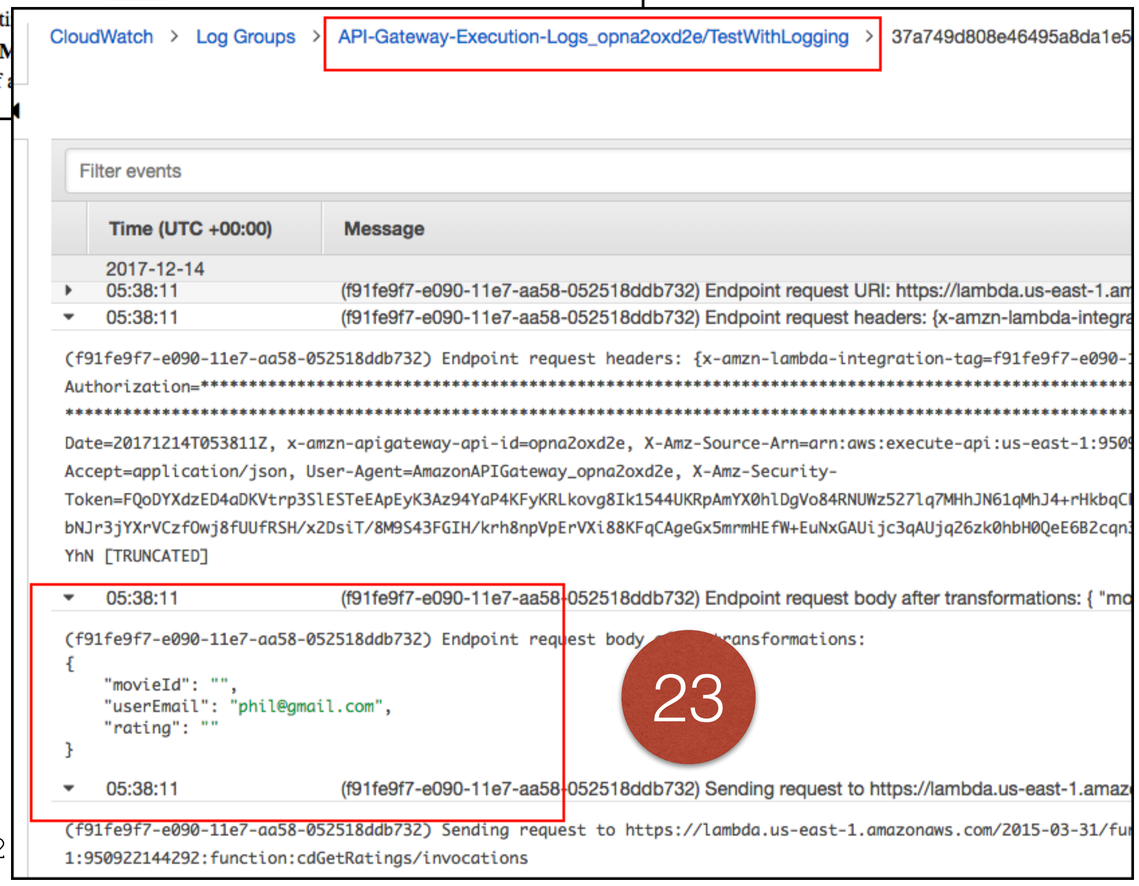
Bind SQS Queue to SNS Topic as Subscriber

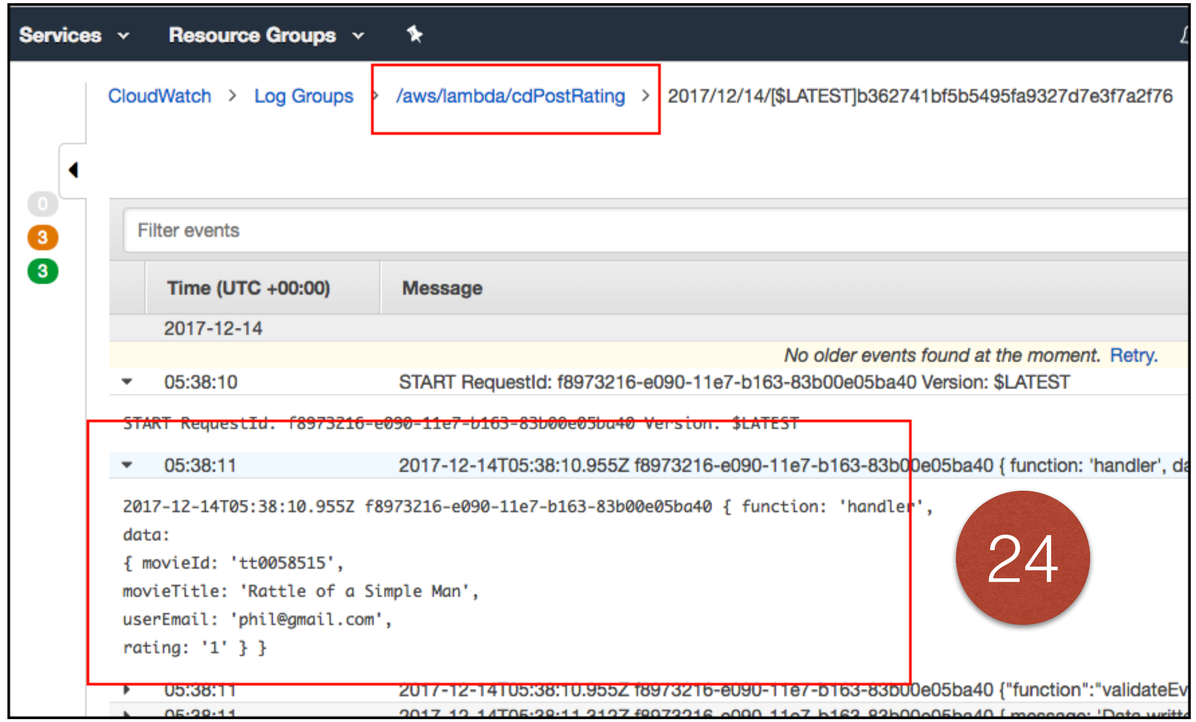
20



Bind SQS Queue to SNS Topic as Subscriber







Day 2 Evaluation

Please take the Day 2 Evaluation:

https://www.surveymonkey.com/r/CK8K6BJ