**Lesson 9 Demo 3**

**Create a Serverless Webpage**

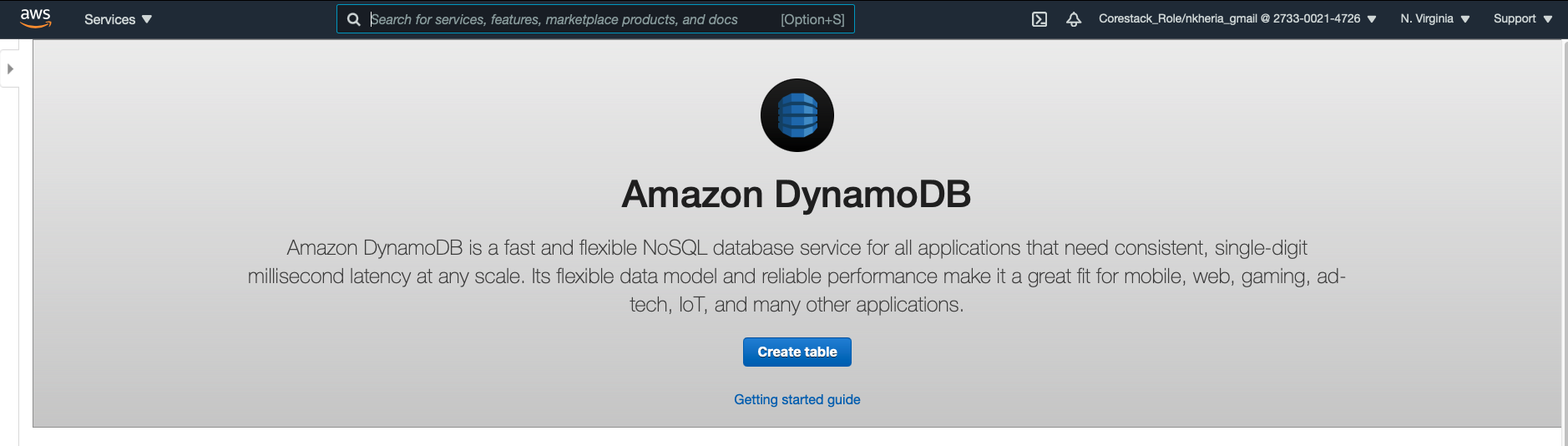


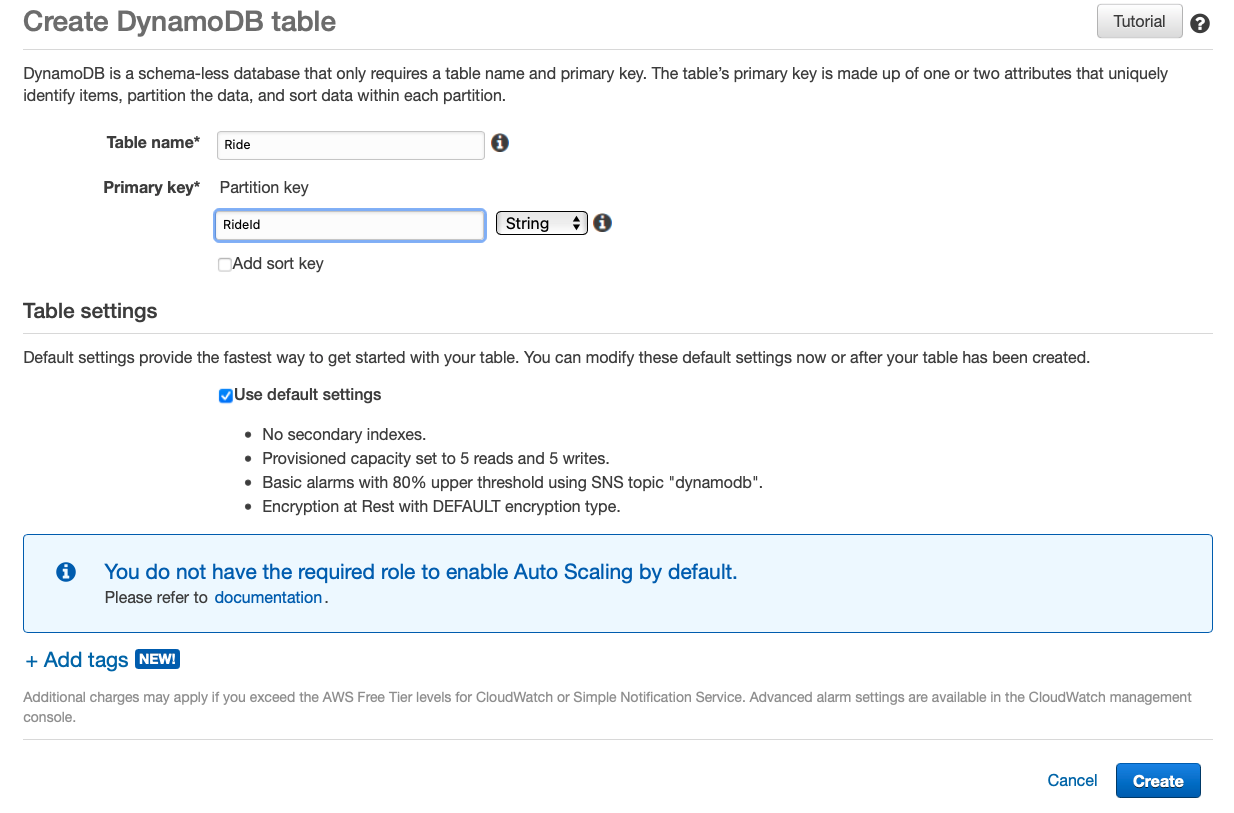
Steps to be followed:

1. Create an Amazon DynamoDB Table
2. Create IAM Role
3. Create lambda function for handling requests
4. Validate you setup

**Step 1. Create an Amazon DynamoDB Table**

1. Use the Amazon DynamoDB console to create a new DynamoDB table.

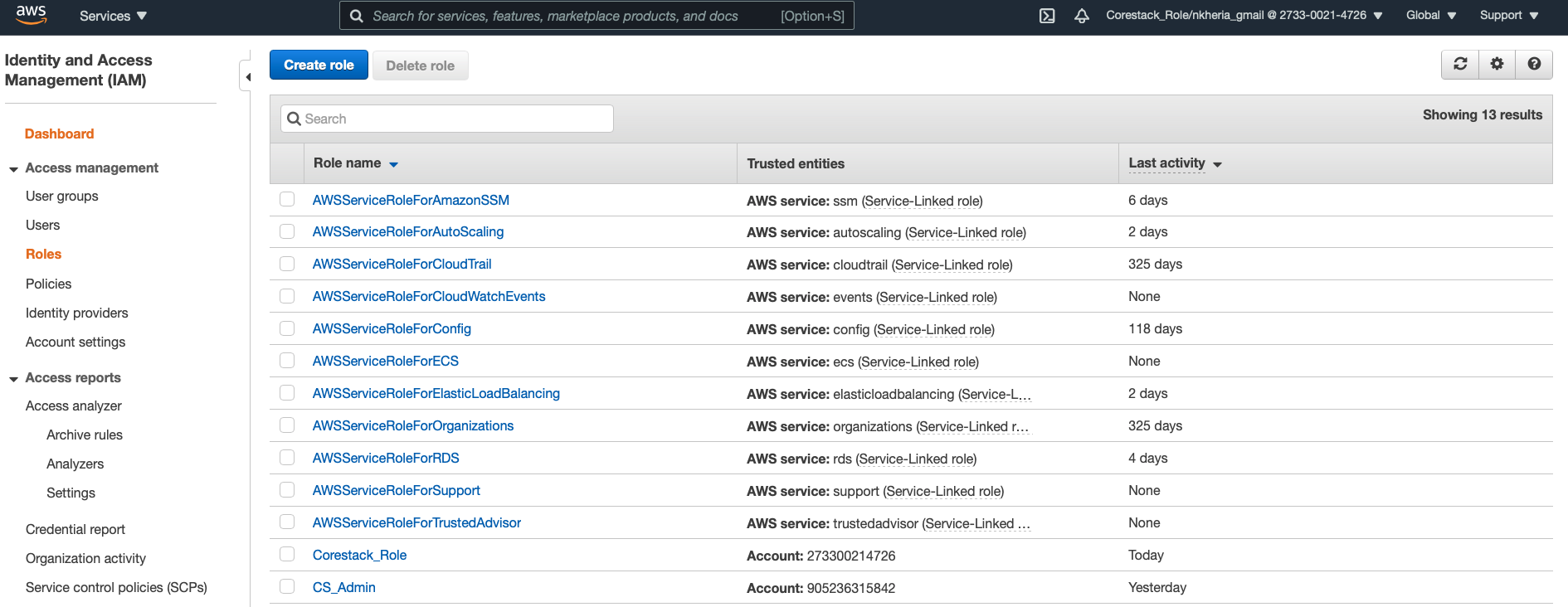




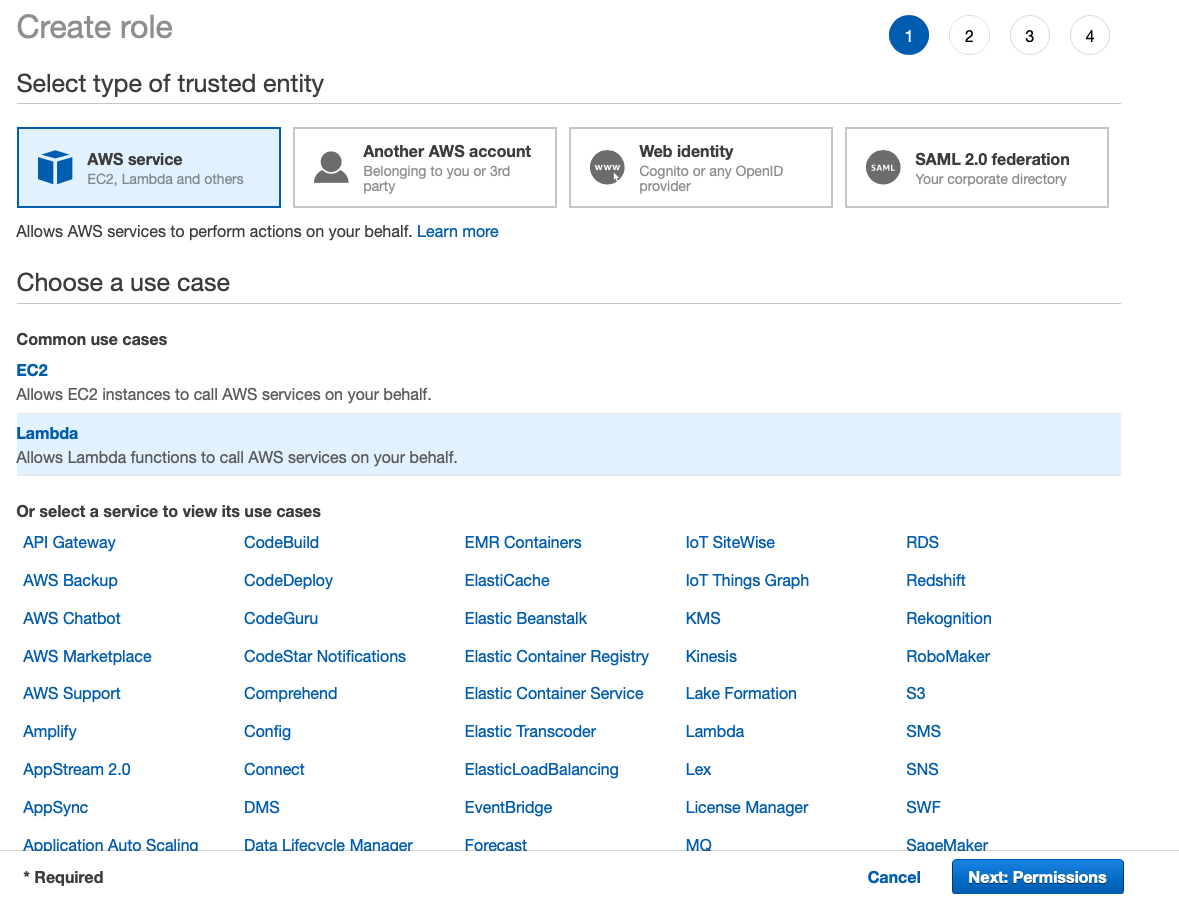
1. Enter Rides for the Table name. This field is case sensitive.
2. Enter RideId for the Partition key and select String for the key type. This field is case sensitive.
3. Check the Use default settings box and choose Create.
4. Scroll to the bottom of the Overview section of your new table and note the ARN. You will use this in the next section.

**Step 2. Create IAM Role**

1. From the AWS Management Console, click on Services and then select IAM in the Security, Identity & Compliance section.
2. Select Roles in the left navigation bar and then choose Create New Role.

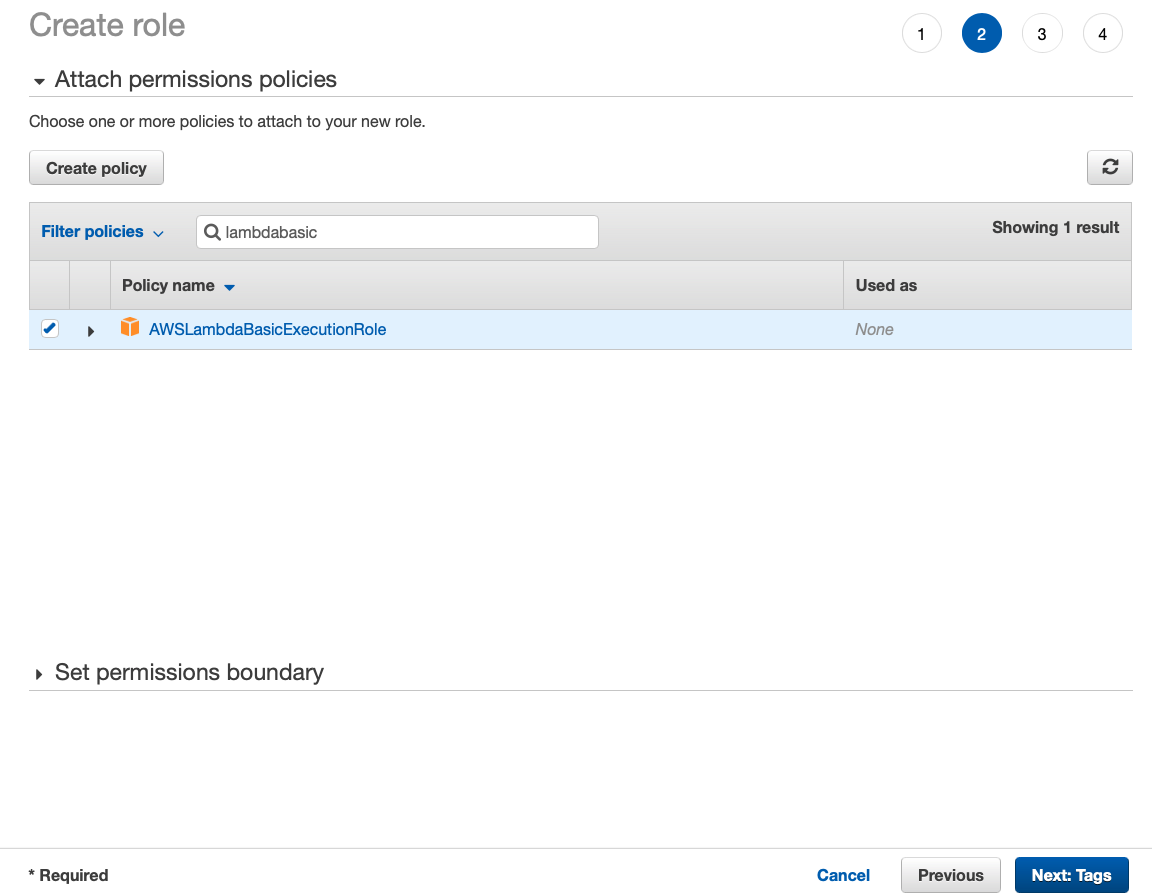


1. Select Lambda for the role type from the AWS service group, then click Next: Permissions.

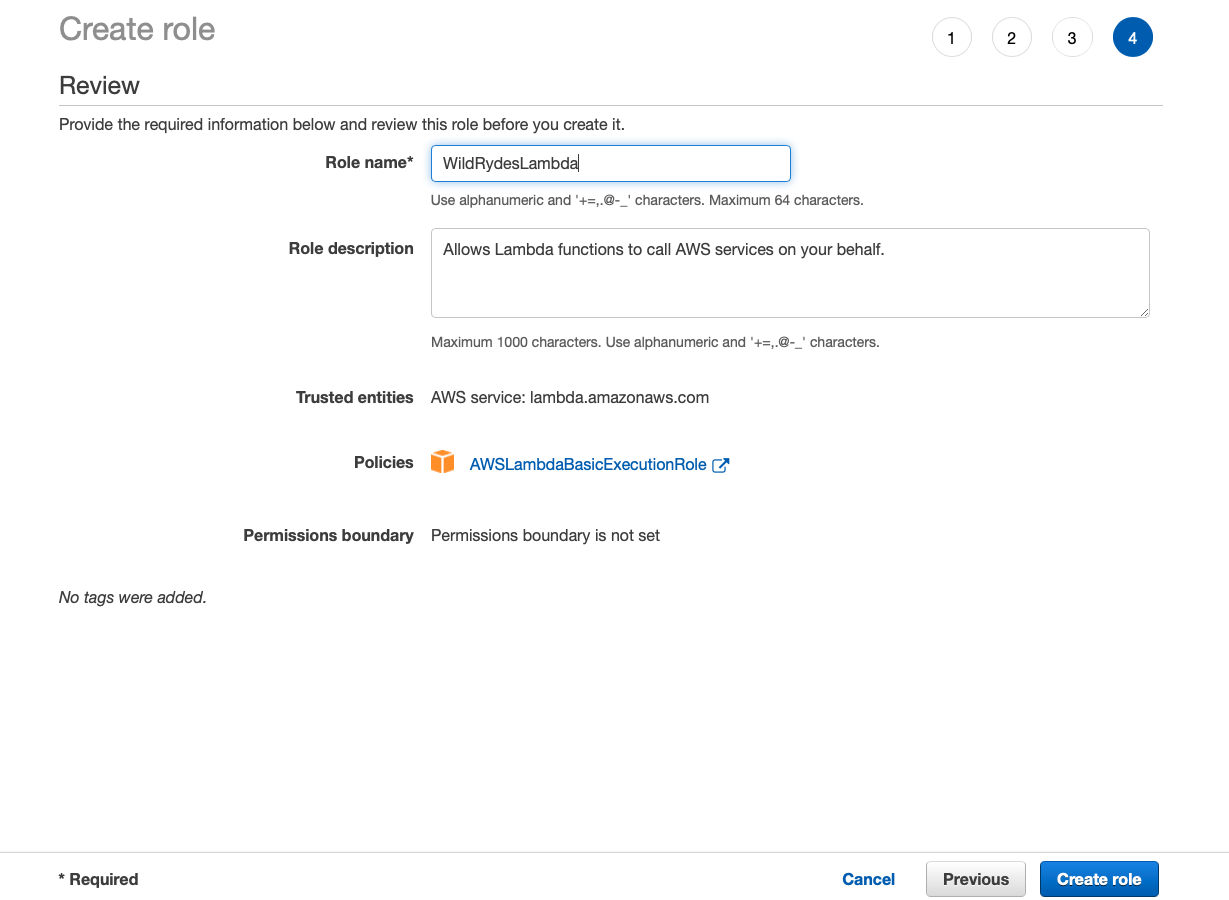




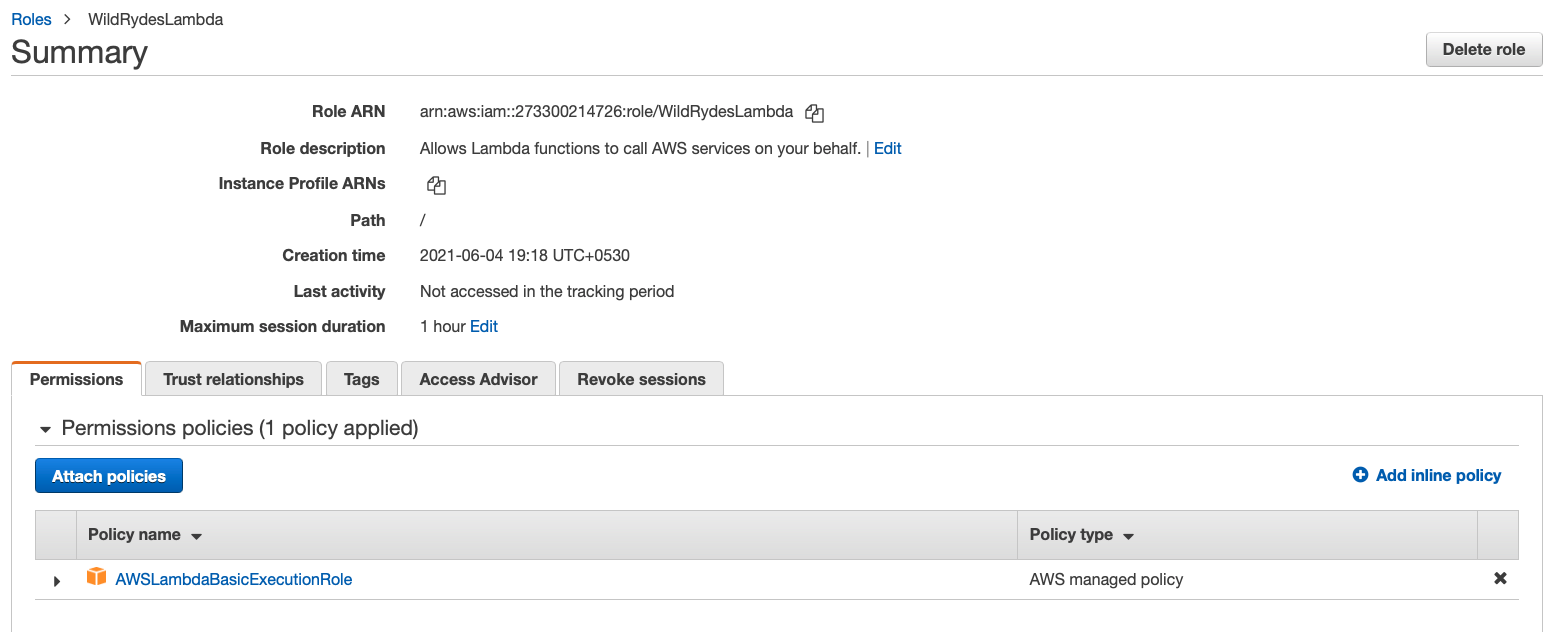
1. Begin typing *AWSLambdaBasicExecutionRole*in the Filter text box and check the box next to that role.



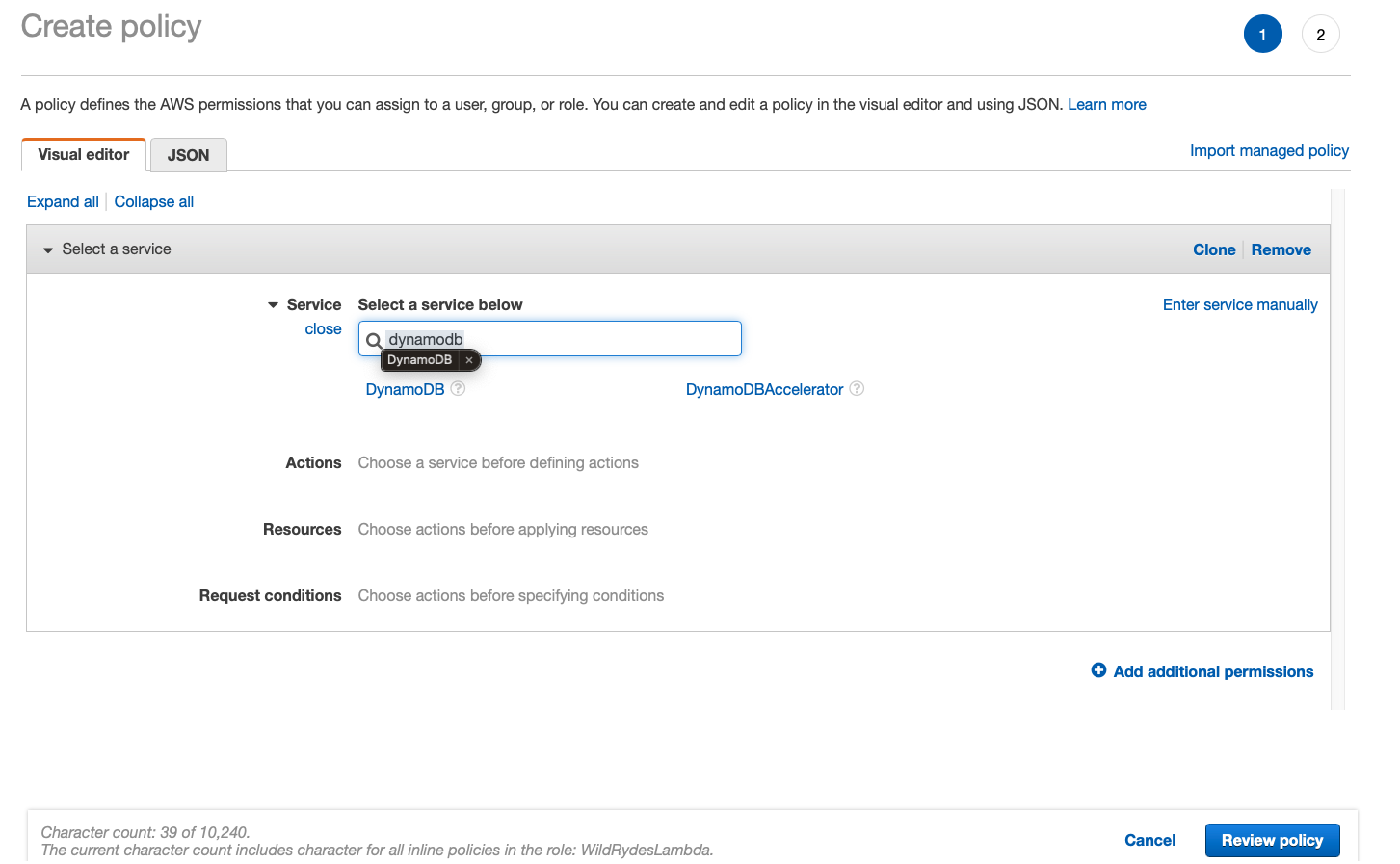
1. Choose Next Step.
2. Enter *WildRydesLambda*for the Role Name.



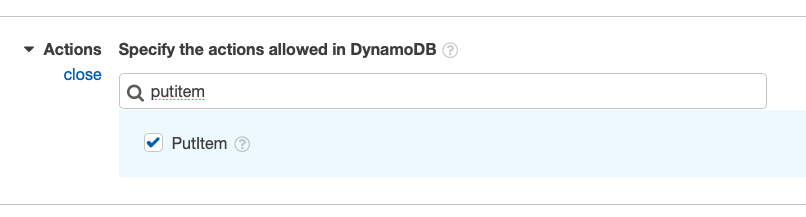
1. Choose Create Role.
2. Type *WildRydesLambda*into the filter box on the Roles page and choose the role you just created.
3. On the Permissions tab, choose the Add inline policy link in the lower right corner to create a new inline policy.



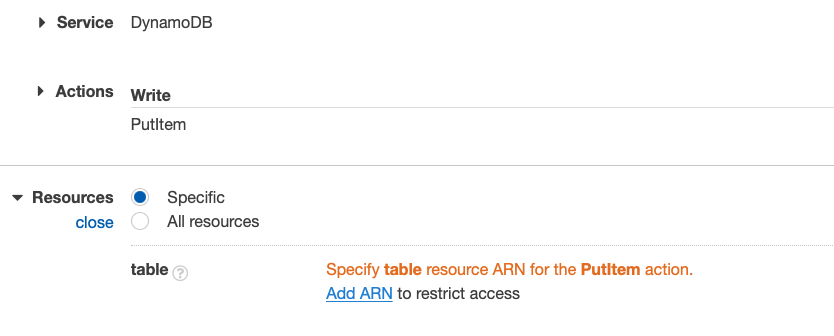
1. Select Choose a service.
2. Begin typing *DynamoDB*into the search box labelled Find a service and select DynamoDB when it appears.



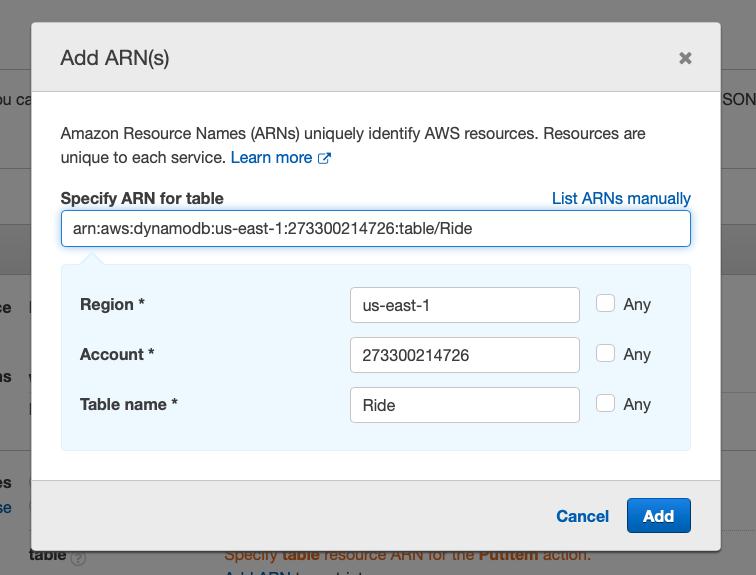
1. Choose Select actions.
2. Begin typing *PutItem*into the search box labelled Filter actions and check the box next to PutItem when it appears.



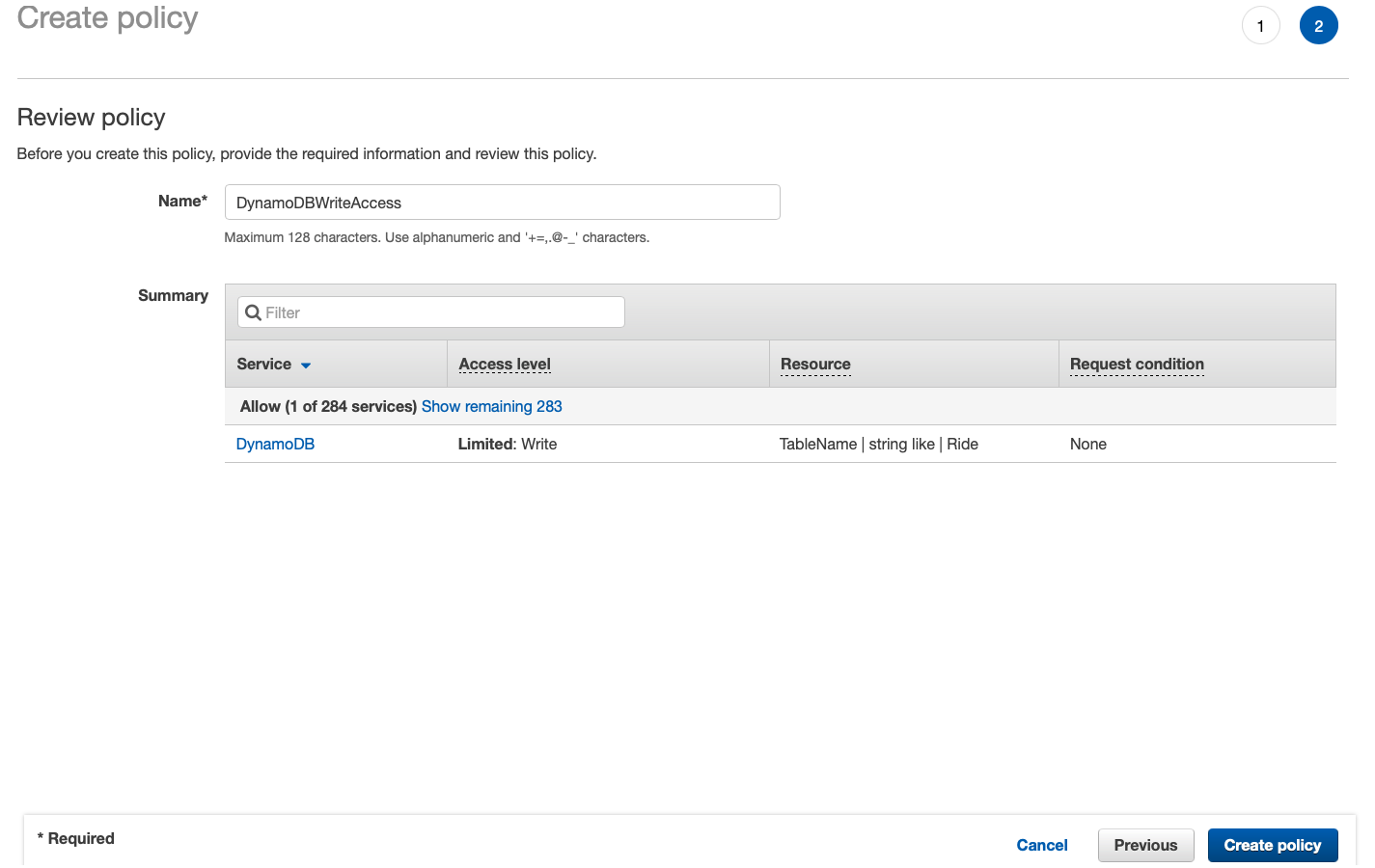
1. Select the Resources section.
2. With the Specific option selected, choose the Add ARN link in the table section.



1. Paste the ARN of the table you created in the previous section in the Specify ARN for table field, and choose Add.



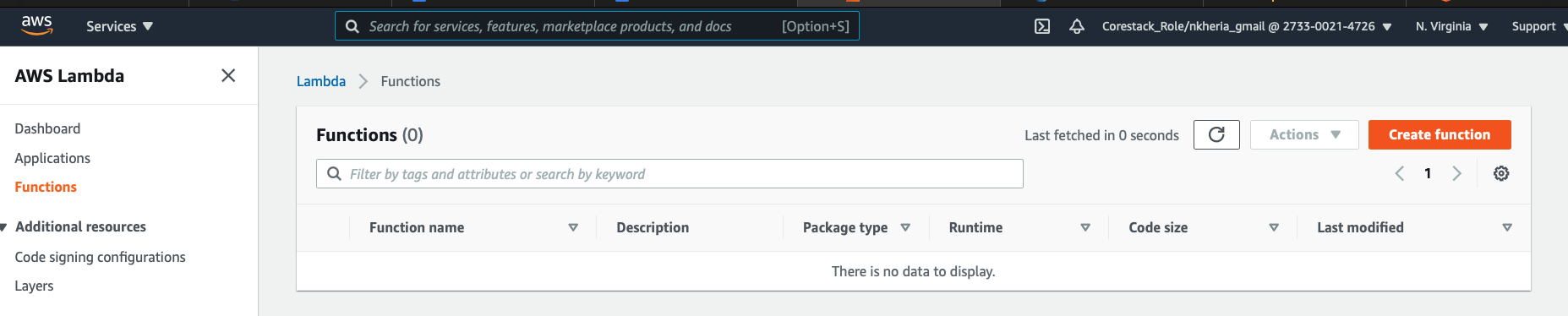
1. Choose Review Policy.
2. Enter *DynamoDBWriteAccess*for the policy name and choose Create policy.



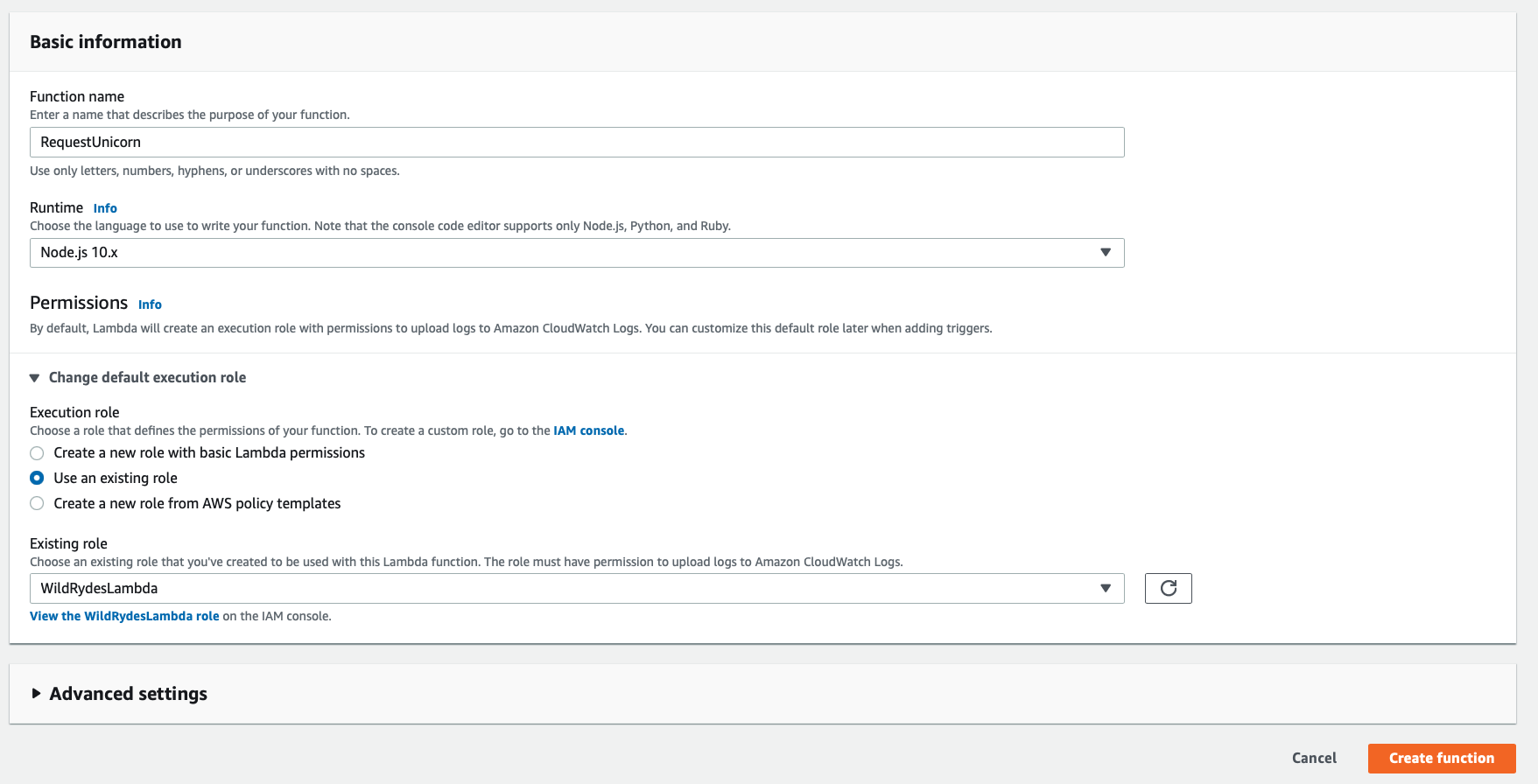
**Step 3. Create lambda function for handling requests**

Make sure to configure your function to use the *WildRydesLambda*IAM role you created in the previous section.

1. Choose Services then select Lambda in the Compute section.
2. Click Create function



1. Keep the default Author from scratch card selected.
2. Enter *RequestUnicorn*in the Name field.
3. Select Node.js.10.x for the Runtime.
4. Ensure *Choose an existing role*is selected from the Role dropdown.
5. Select *WildRydesLambda*from the Existing Role dropdown.
6. Click on Create function.

******

1. Scroll down to the Function code section and replace the existing code in the index.js code editor with the contents of [***requestUnicorn.js***](https://webapp.serverlessworkshops.io/serverlessbackend/lambda/requestUnicorn.js)***.***
2. Click "Save" in the upper right corner of the page.

**Step 4. Validate you setup**

1. From the main edit screen for your function, select Configure test event from the the Select a test event...dropdown.
2. Keep Create new test event selected.
3. Enter *TestRequestEvent* in the Event name field
4. Copy and paste the following test event into the editor:

{

"path": "/ride",

"httpMethod": "POST",

"headers": {

"Accept": "\*/\*",

"Authorization": "eyJraWQiOiJLTzRVMWZs",

"content-type": "application/json; charset=UTF-8"

},

"queryStringParameters": null,

"pathParameters": null,

"requestContext": {

"authorizer": {

"claims": {

"cognito:username": "the\_username"

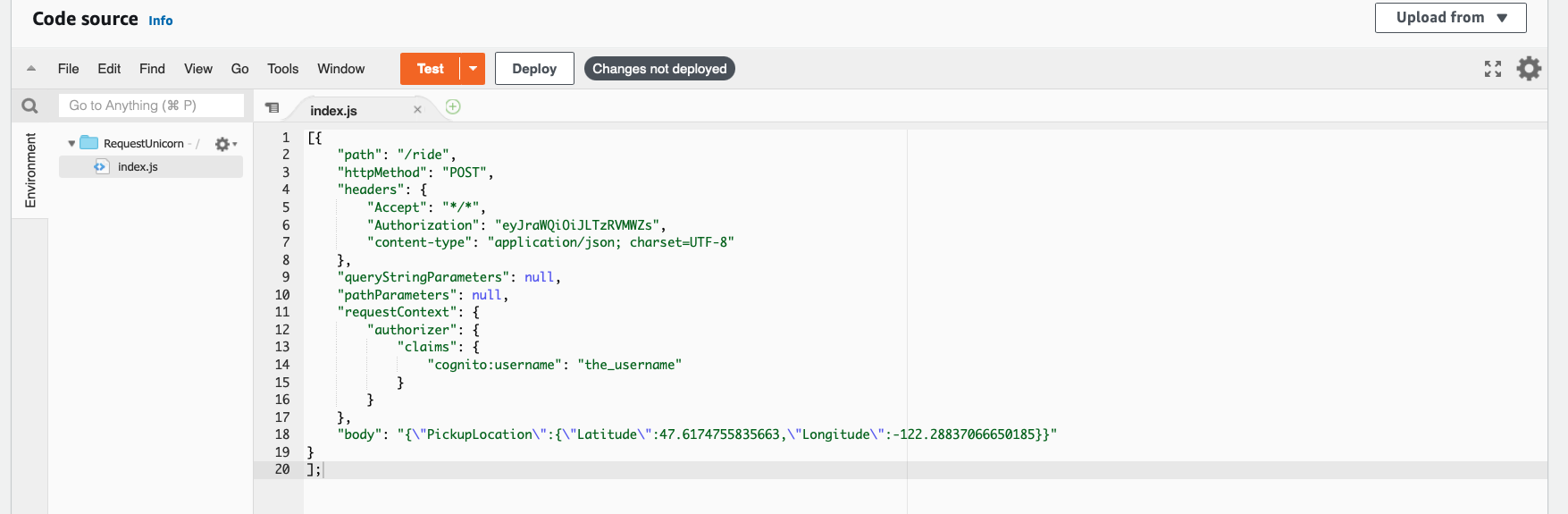
}

}

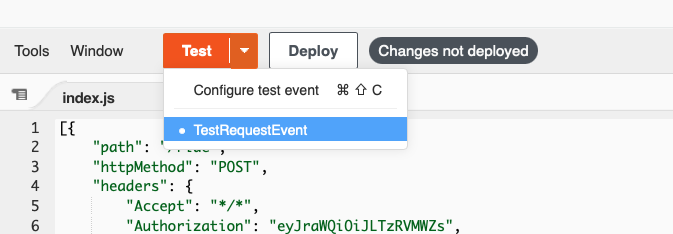
},

"body": "{\"PickupLocation\":{\"Latitude\":47.6174755835663,\"Longitude\":-122.28837066650185}}"

}



1. Click Create.
2. On the main function edit screen click Test with *TestRequestEvent*selected in the dropdown.



1. Scroll to the top of the page and expand the Details section of the Execution result section.
2. Verify that the execution succeeded and that the function result looks like the following:

{

"statusCode": 201,

"body": "{\"RideId\":\"SvLnijIAtg6inAFUBRT+Fg==\",\"Unicorn\":{\"Name\":\"Rocinante\",\"Color\":\"Yellow\",\"Gender\":\"Female\"},\"Eta\":\"30 seconds\"}",

"headers": {

"Access-Control-Allow-Origin": "\*"

}

}