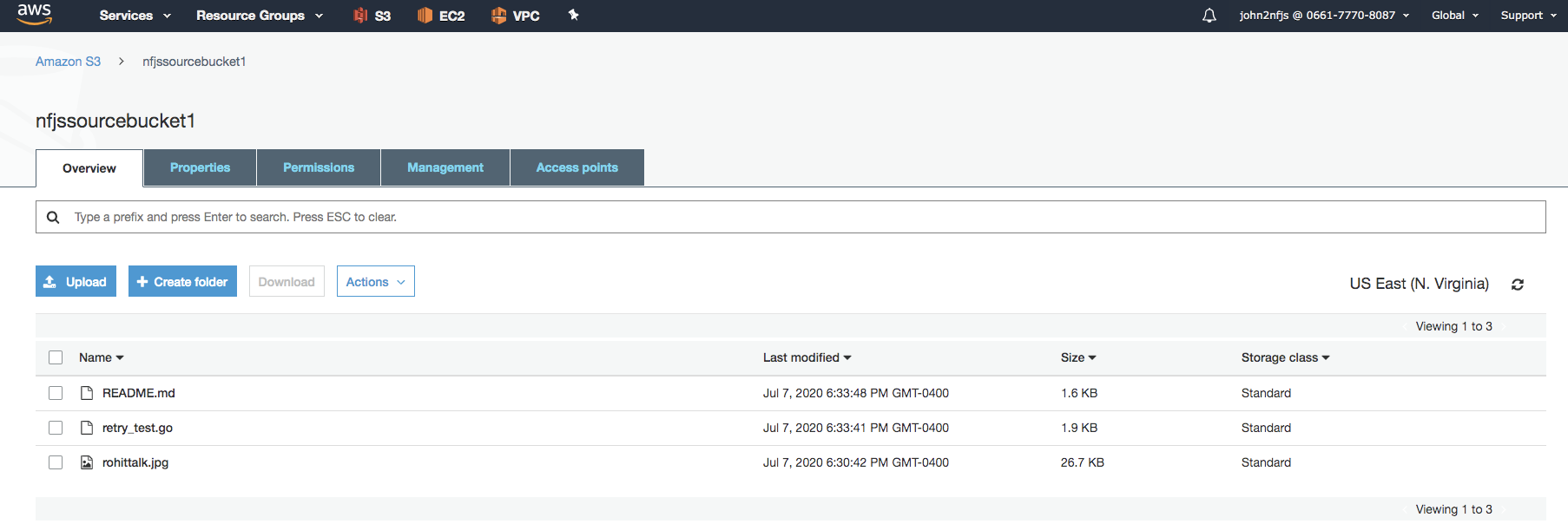
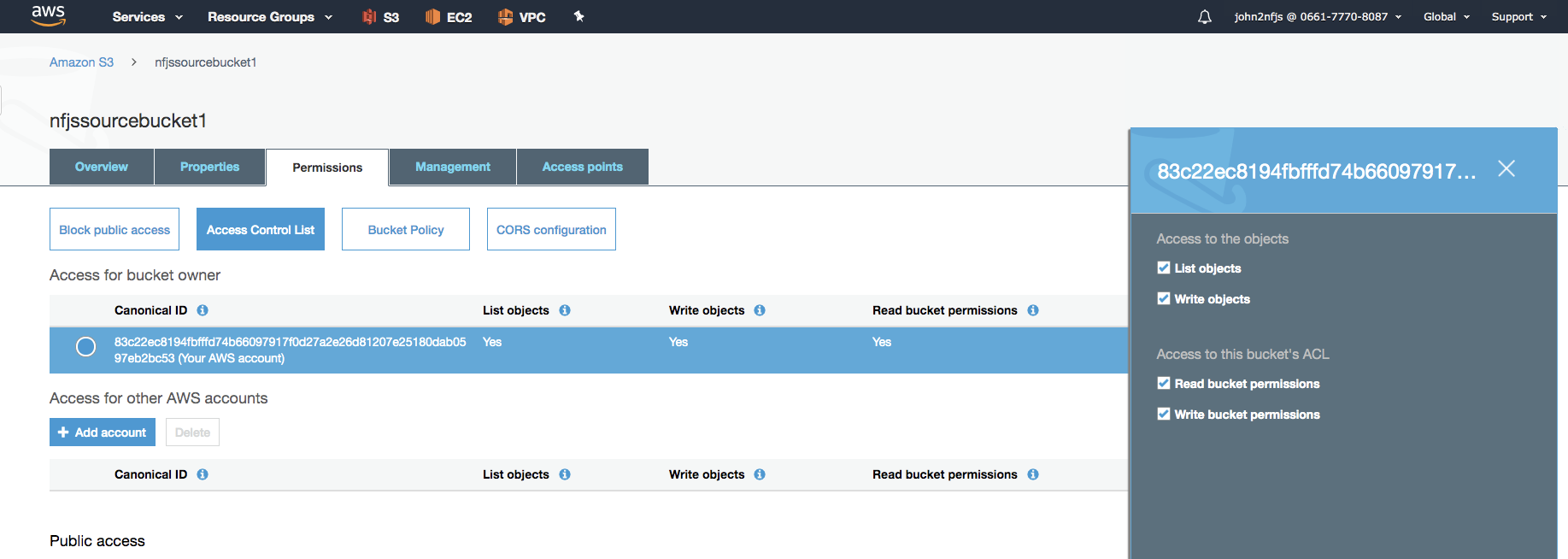
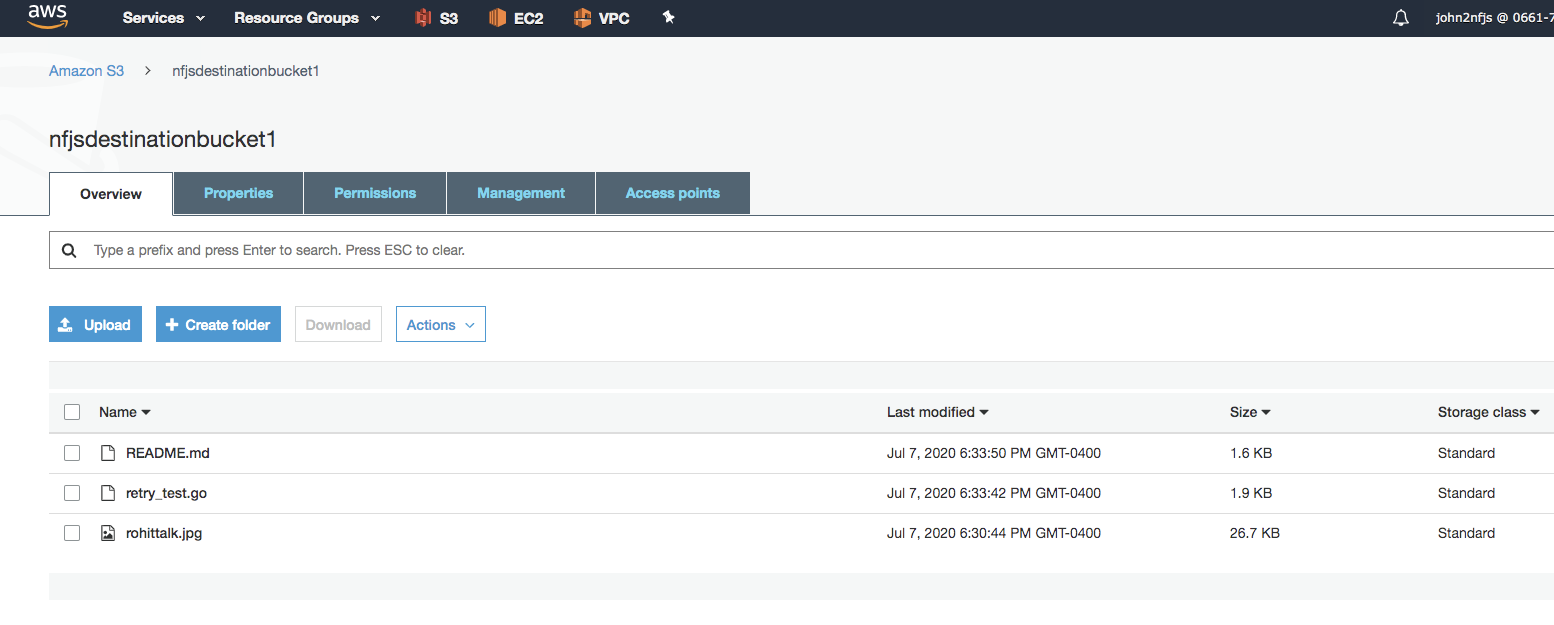
# Amazon Lambda with input from S3

## Steps

### Create two S3 buckets. One for the source and one for the destination.

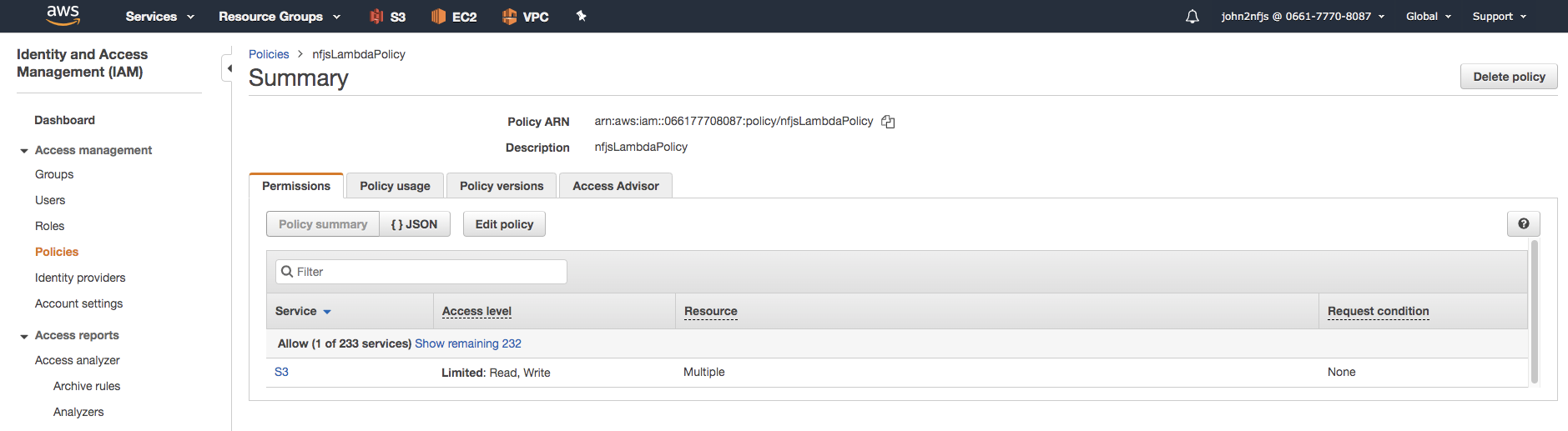






arn:aws:s3:::nfjssourcebucket1

### Create an IAM Policy



{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::nfjssourcebucket1/\*"

},

{

"Sid": "VisualEditor1",

"Effect": "Allow",

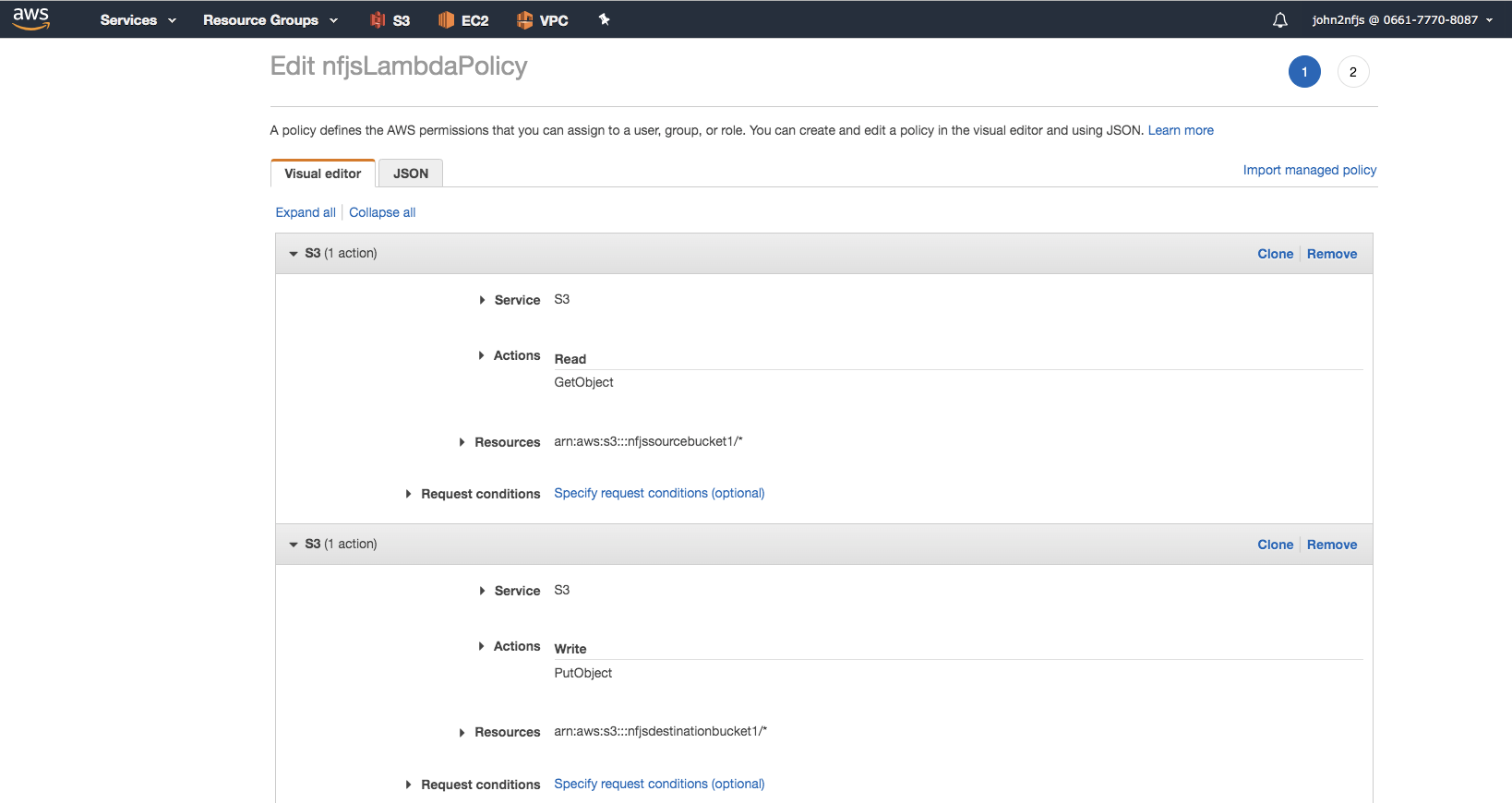
"Action": "s3:PutObject",

"Resource": "arn:aws:s3:::nfjsdestinationbucket1/\*"

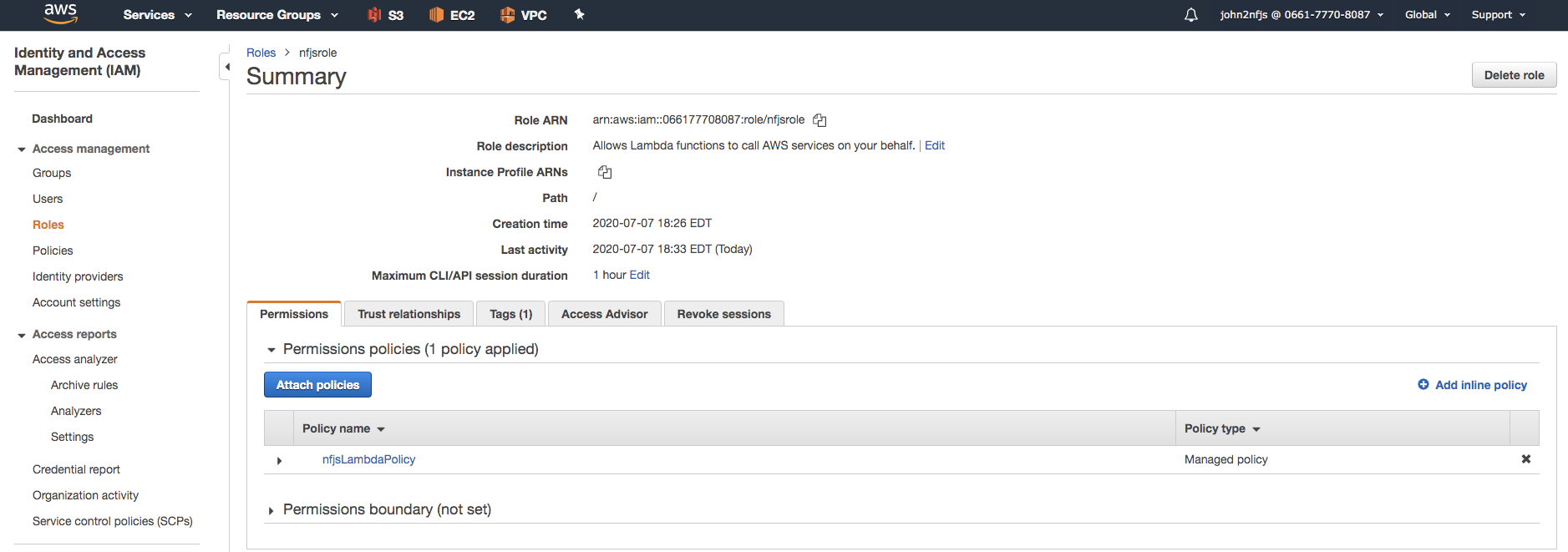
}

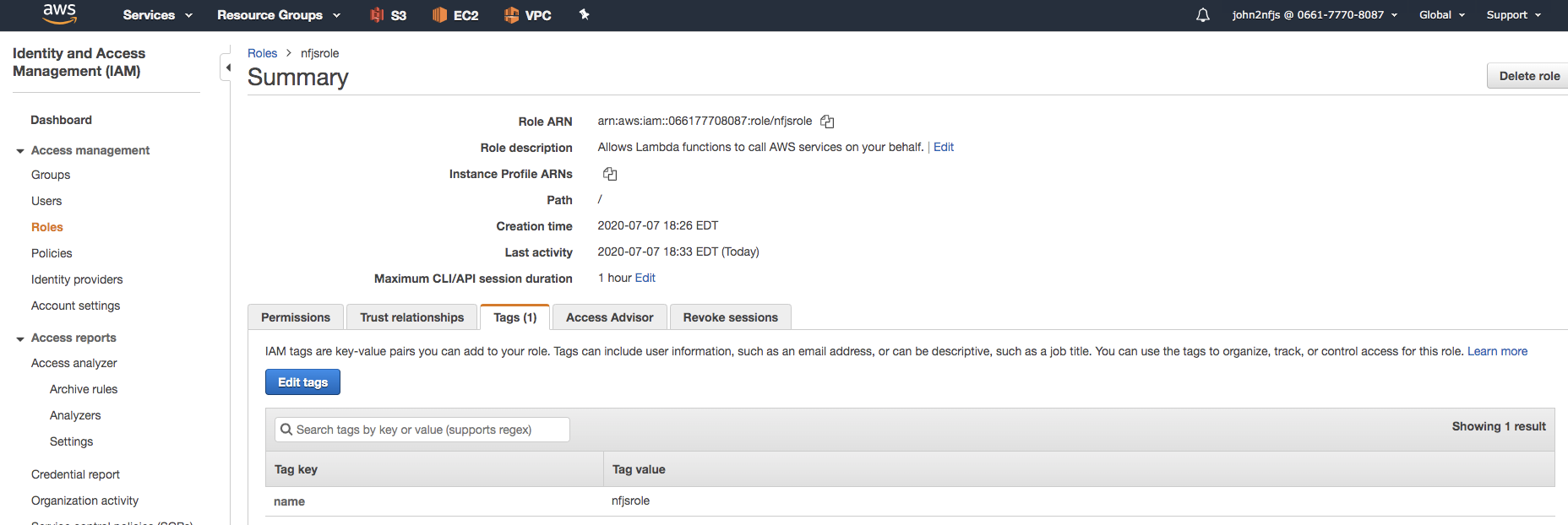
]

}



### Create an IAM Role



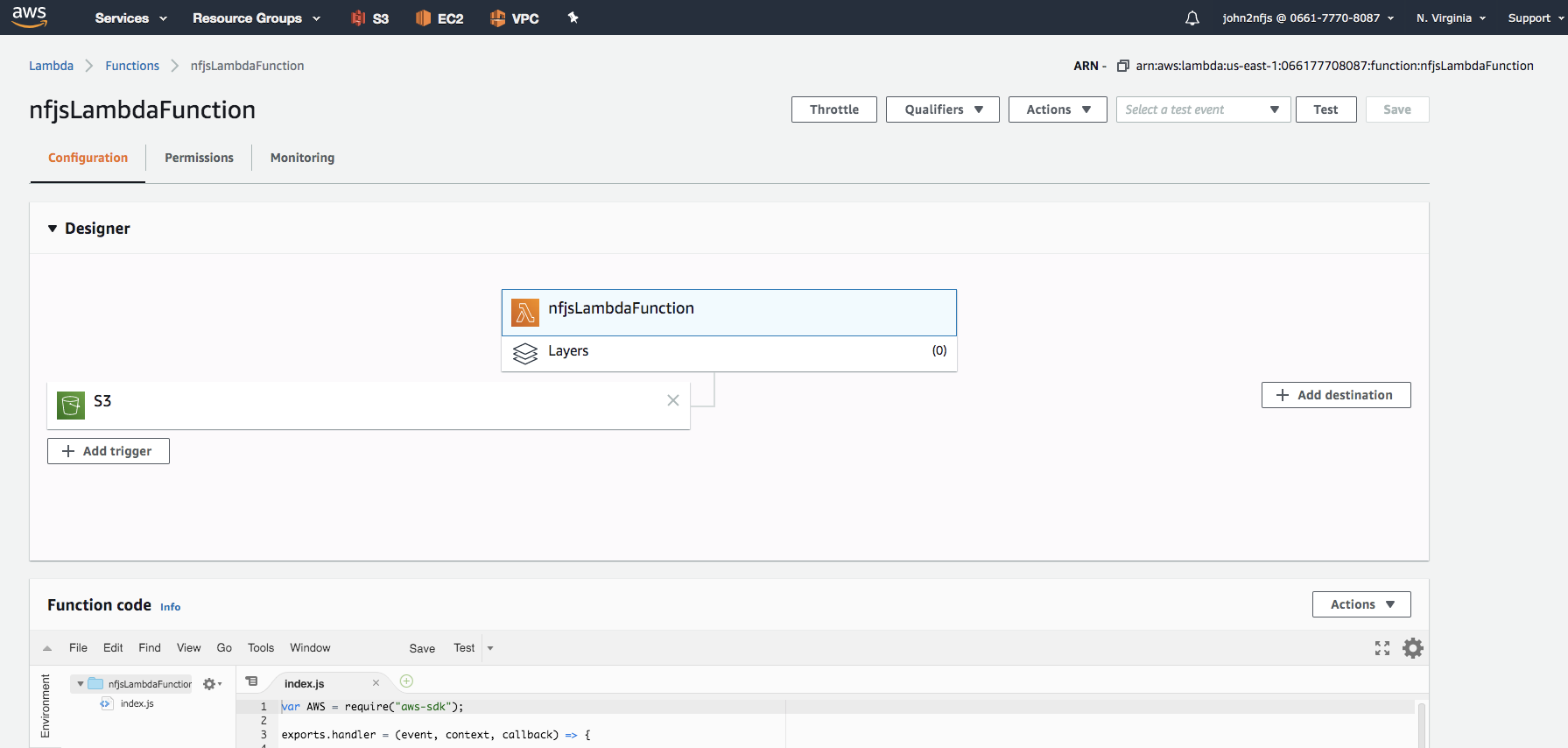


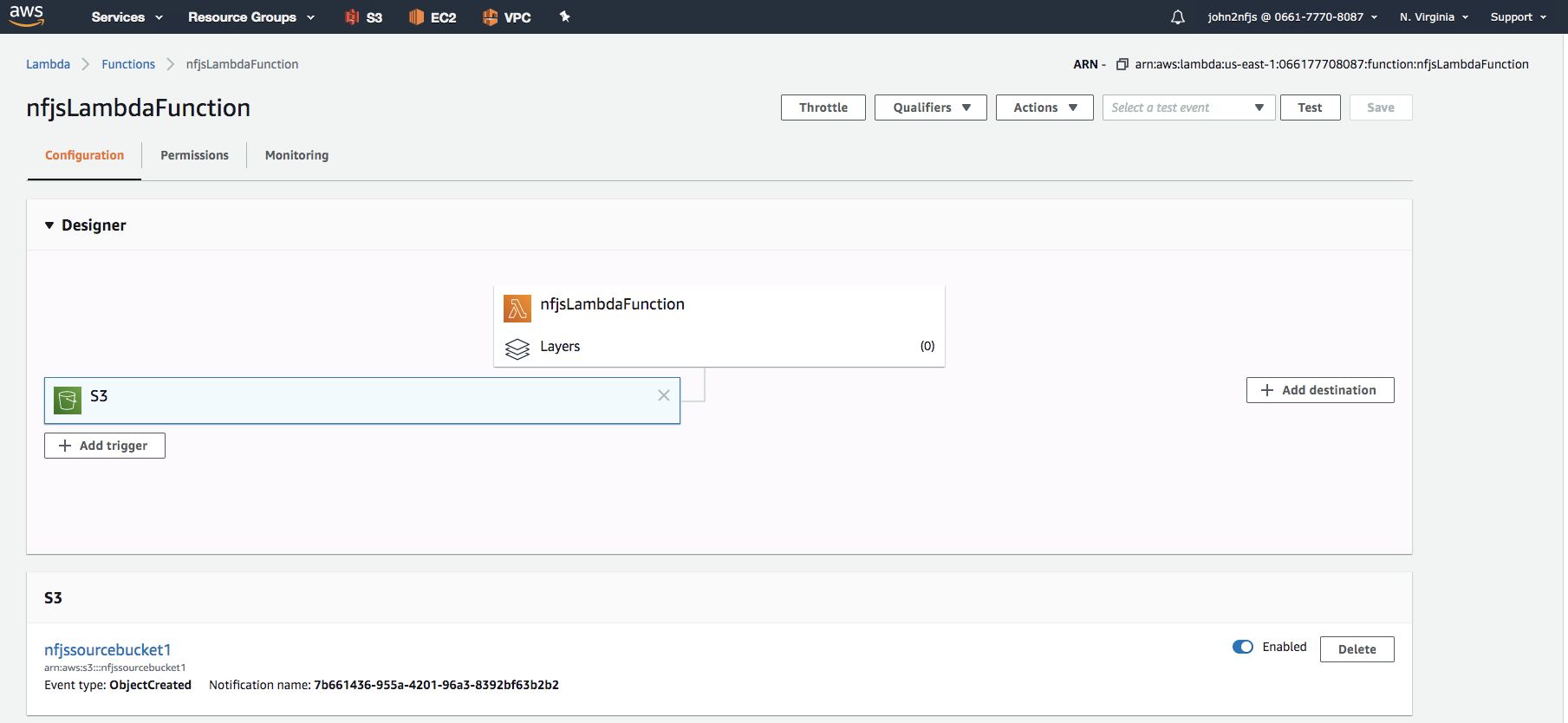
### Create a Lambda function to copy the object from one bucket to another bucket.

NodeJs is Runtime

Role: Pick the role just created

### 





Node.js

var AWS = require("aws-sdk");

exports.handler = (event, context, callback) => {

var s3 = new AWS.S3();

var sourceBucket = "**nfjssourcebucket1**";

var destinationBucket = "**nfjsdestinationbucket1**";

var objectKey = event.Records[0].s3.object.key;

var copySource = encodeURI(sourceBucket + "/" + objectKey);

var copyParams = { Bucket: destinationBucket, CopySource: copySource, Key: objectKey };

s3.copyObject(copyParams, function(err, data) {

if (err) {

console.log(err, err.stack);

} else {

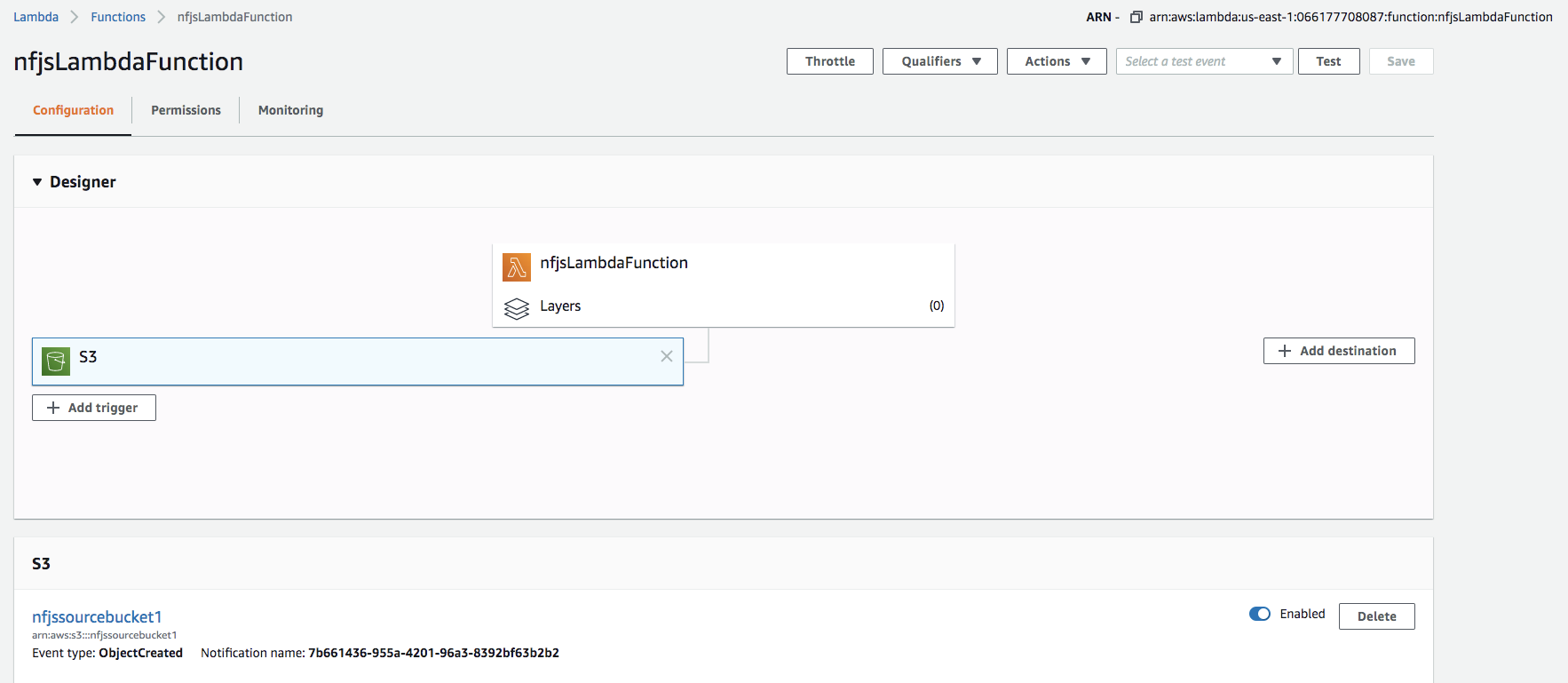
console.log("S3 object copy successful.");

}

});

};

Add Trigger to Lambda function: S3 source bucket



### Test the Lambda Function.

Upload files to Source and test if files are copied to Destination.