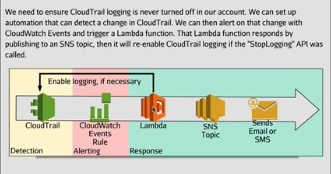
CloudTrail Documentation

What is CloudTrail? CloudTrail is a monitoring tool in AWS that logs all public plane API calls in your AWS account to a S3 bucket. It can provide valuable information when assessing incidents and it is best practice to keep CloudTrail logging at all time. Let us pretend that someone in your company decided to mess around in the companies AWS account and they started deleting resources. Maybe these resource are important like full S3 buckets or KMS encryption keys. How would you figure out who was being malicious? Well to answer that you would check your CloudTrail log files.

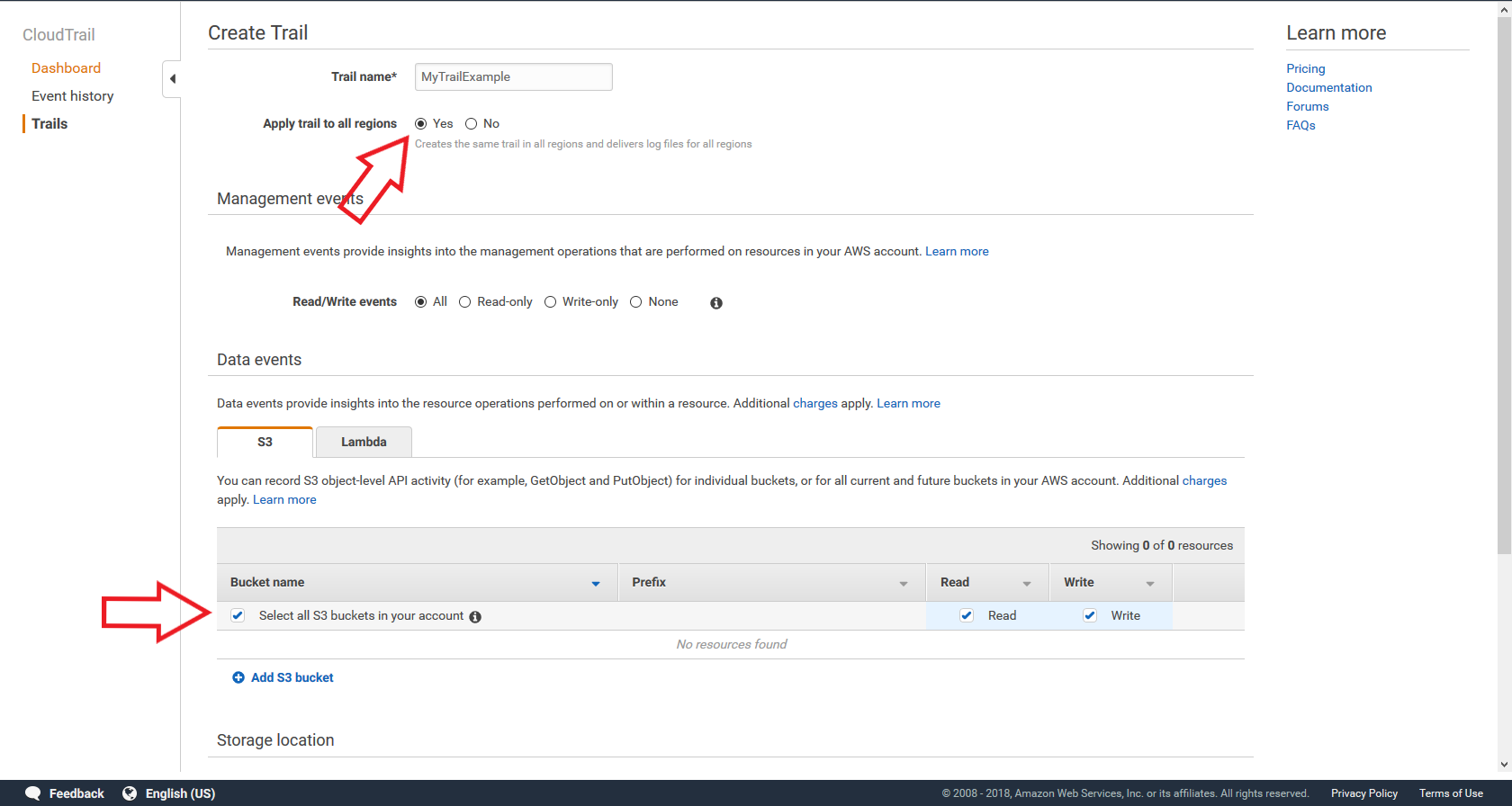
Why would someone turn off CloudTrail? Maybe the person who is being malicious really knows what they are doing. Maybe they know how not to be caught, or maybe you just hired a new guy and he accidentally disables CloudTrail. Whatever the reason, CloudTrail logging of AWS is a security best practice and if you are in any really world environment you should always have one trail logging your account.

This guide will show you how to setup CloudTrail so that if a trail is disabled it will invoke a Lambda function and send your person an email so you can check out the incident and see who did it.

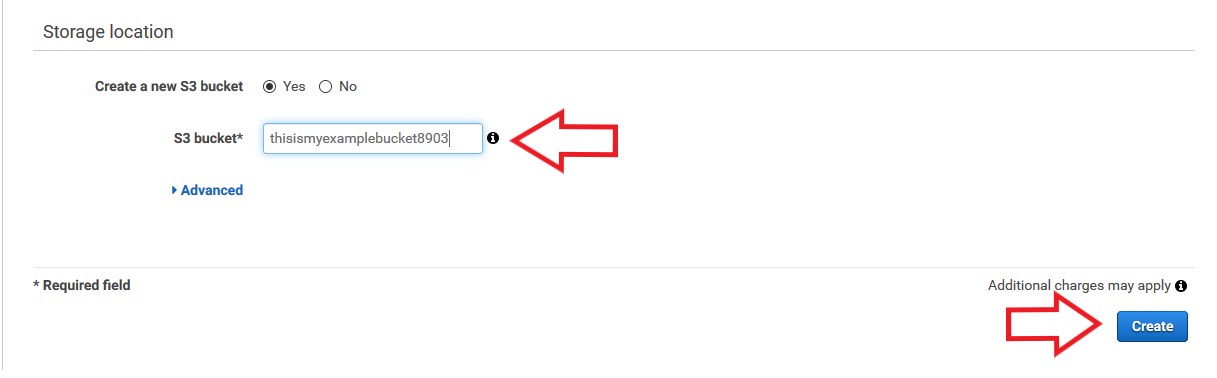


**Step 1 Setting Up a Trail:**

Sign into your AWS account and go to CloudTrail. Click on view trails and locate the blue create trail button. Name the trail whatever you want, make sure that apply trail to all regions is set to yes, and enable data events for S3 and Lambda.

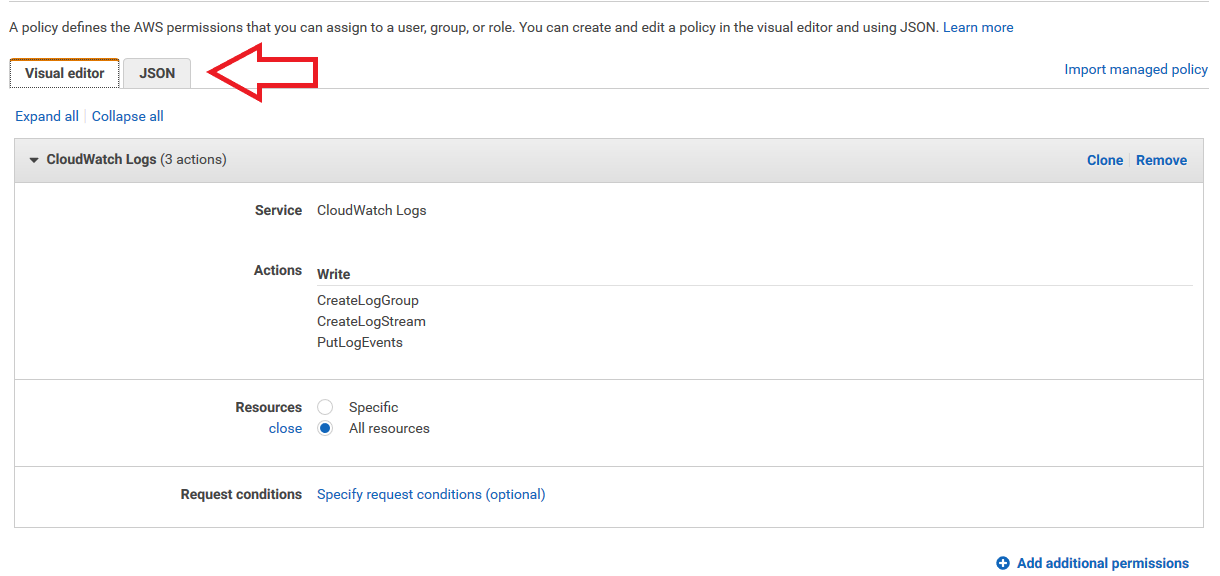


Next scroll down to storage location and either create a new S3 bucket or choose one already in your account. Then click create and your trail will be created.

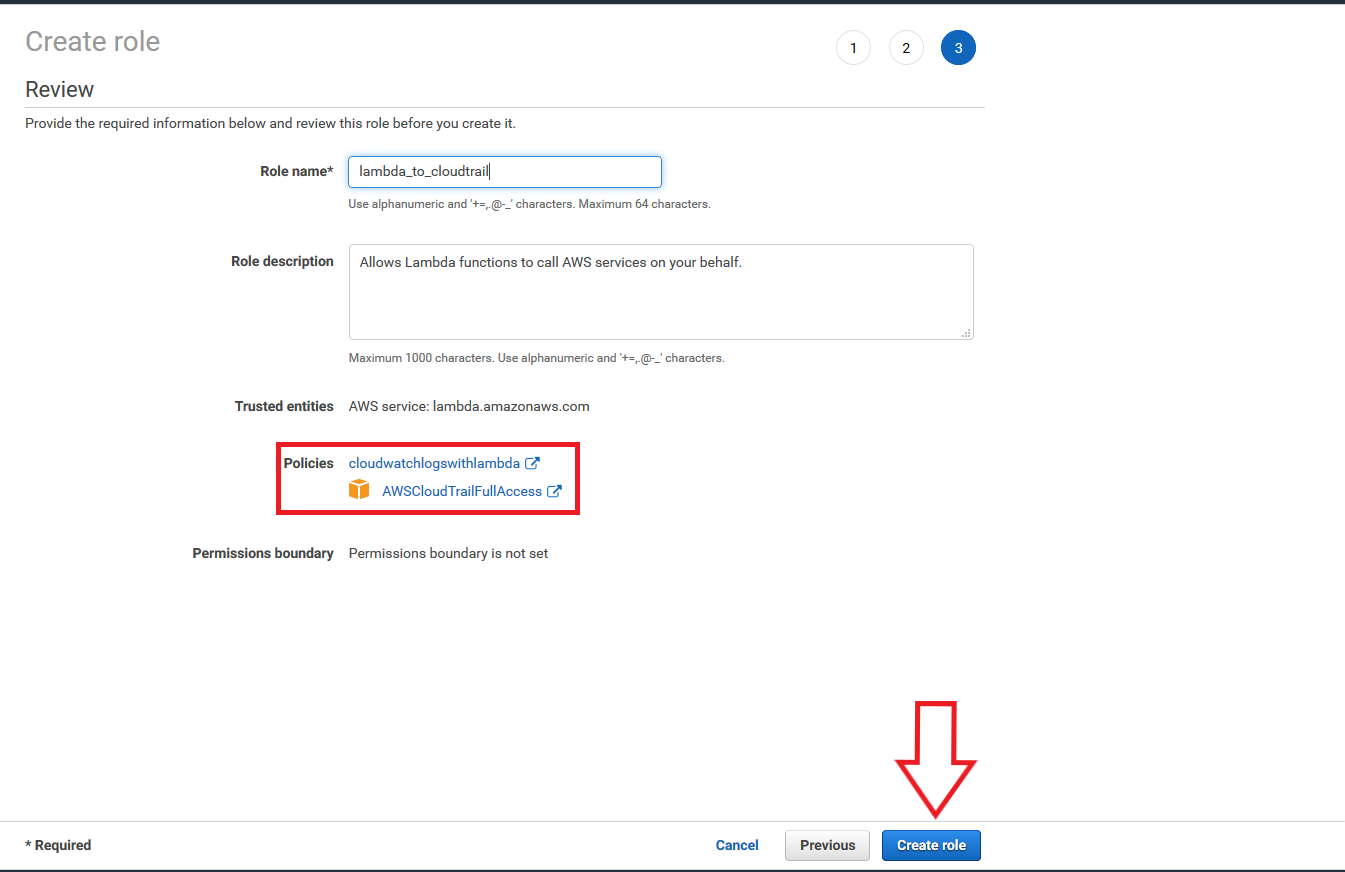


**Step 2 Create a Lambda Role:**

Go to IAM and click on roles and click on create role. This will be a Lambda role so choose Lambda as the service using the role and click on next permissions and click create policy. The policy I made for this role is located in my GitHub account in the Lambda policy file. You can either copy based on this screen shot or you can go to the JSON tab highlighted by the red arrow and copy and paste the JSON provided in the GitHub account.

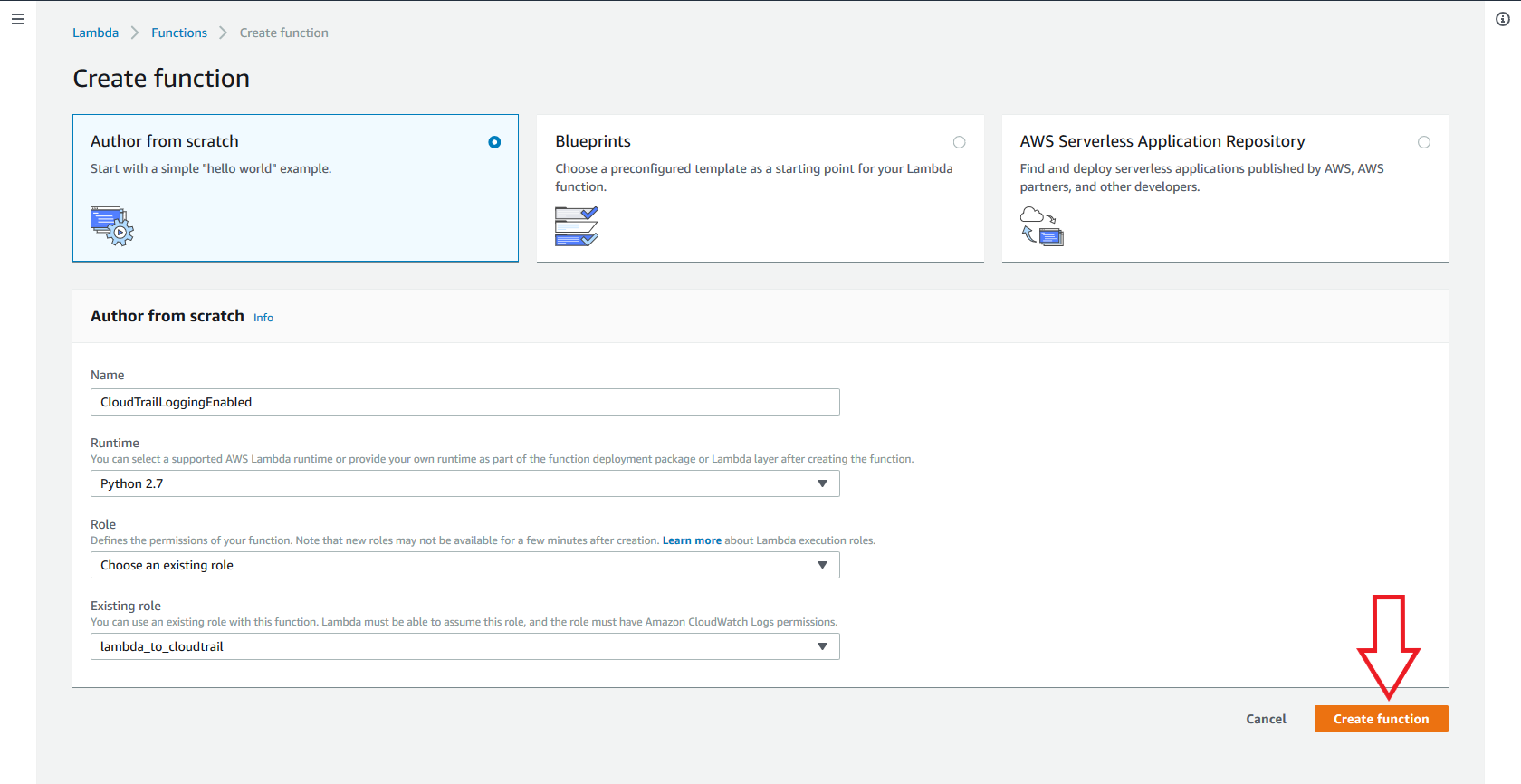


After you finish that click review policy and name the policy. Go back to the roles page and locate the role you just created. Select that policy and locate an AWS made policy titled CloudTrail full access. This policy will allow Lambda to re enable any trail that is turned off. Click next review and name the role whatever you want. Make sure that there are 2 policies attached allowing the correct permissions highlighted by the red box and then click create role.

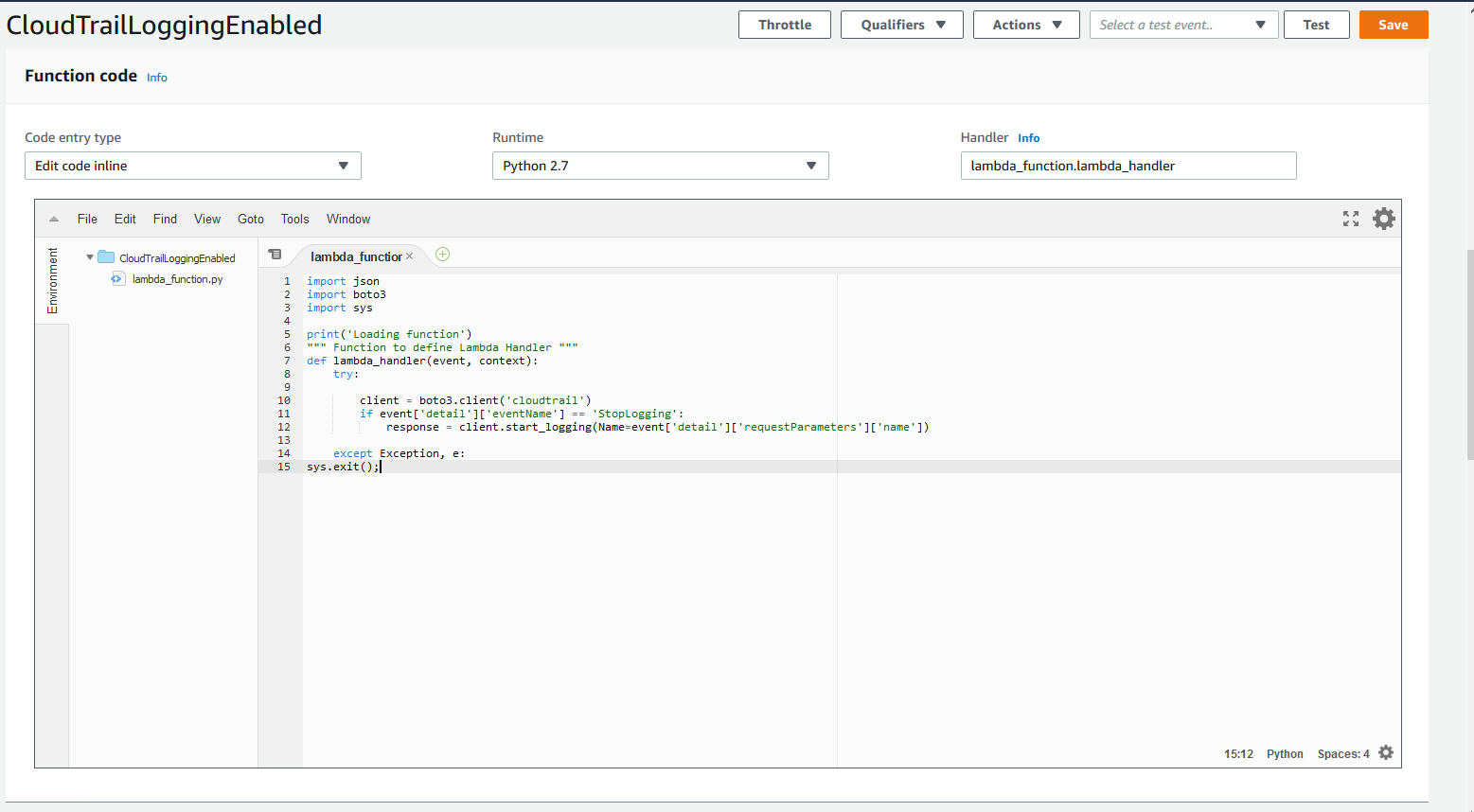


**Step 3 Creating the Lambda Function:**

Go to the Lambda service in AWS and click create a function, author from scratch, name it, select the Python 2.7 for your runtime, choose an existing role and use the role you just created, and click create function.

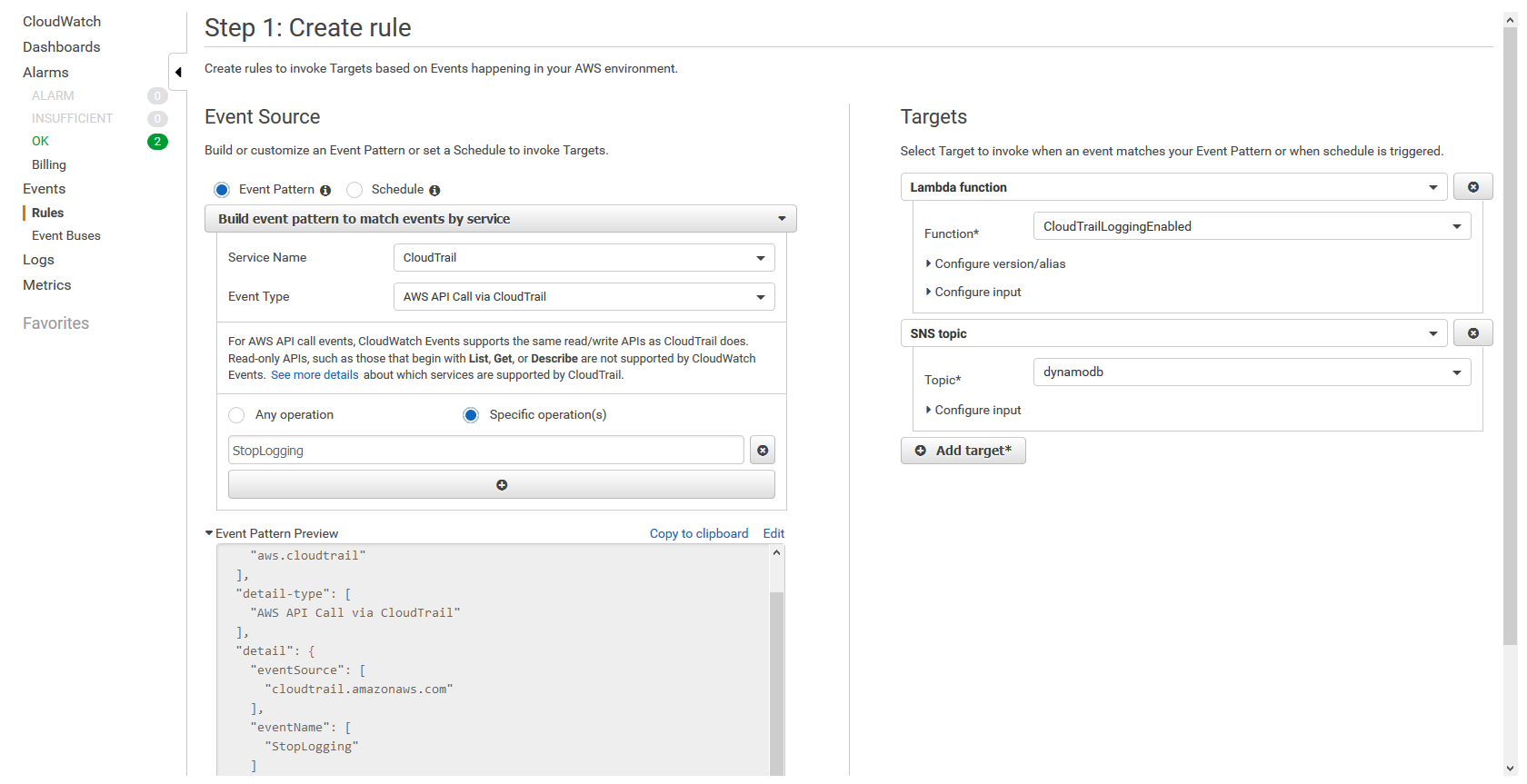


Scroll down to the code section and locate the code in my repository located in a file called CloudTrail. Copy and paste that code into your lambda function and then click on save located in the top right of your screen.



**Step 4 Setting up CloudWatch Events:**

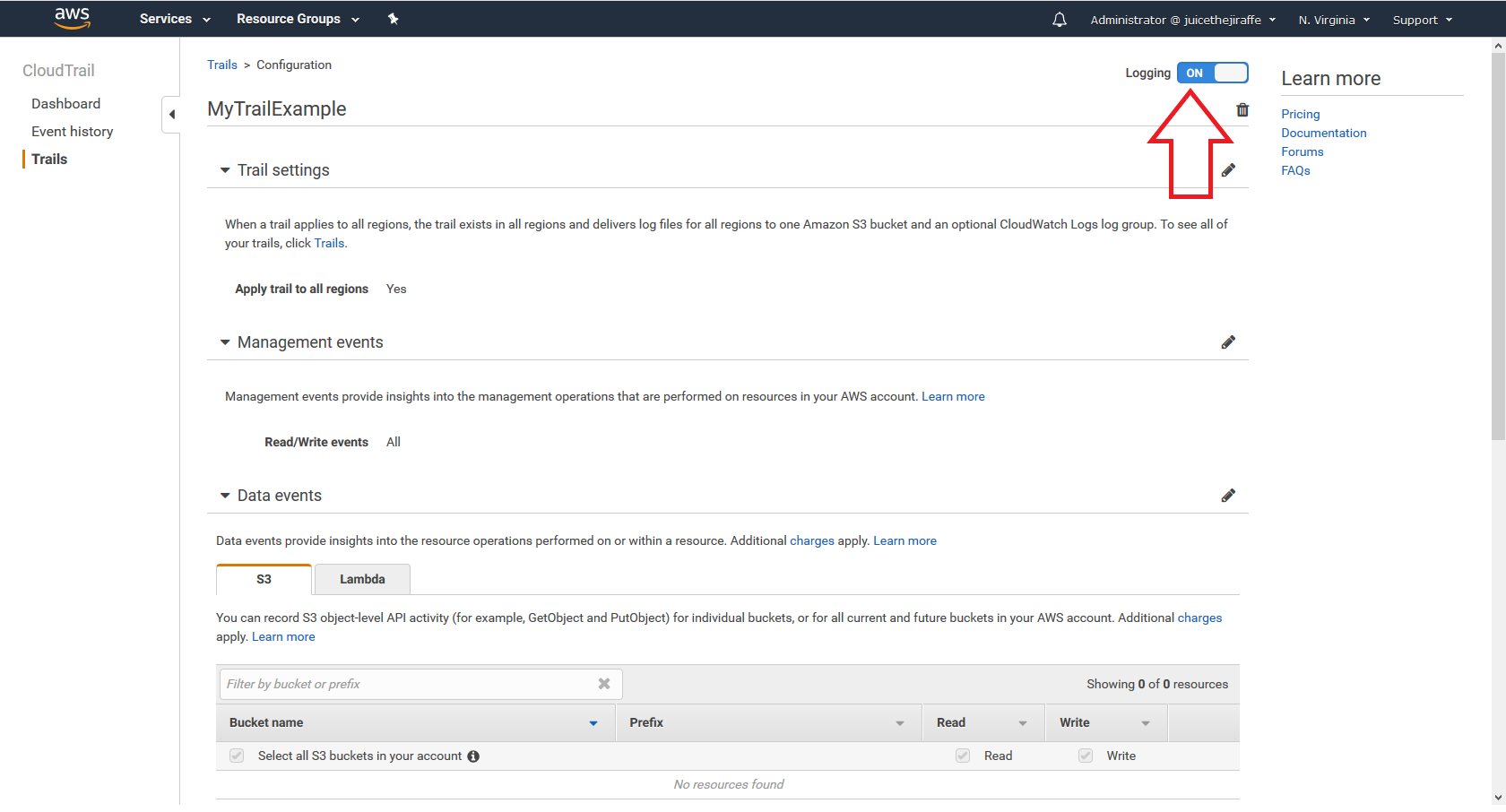
Go to CloudWatch and locate the rules section on the left hand side and click create rule. We want the rule to be based on an event pattern off of CloudTrail and we need to look for API calls against CloudTrail. The API call we will look for is a specific API operation called StopLogging (keep in mind that caps does matter). Now add the Lambda function as a target. You can optionally add another target that is SNS to notify you or your team for when this event is invoked. Click configure details.



Name the rule and give it a description then click create rule.

**Step 5 Test the Rule:**

To test the rule go to the trail you made at the beginning and click on it to go to trail settings. Disable logging by clicking the button at the top right and refresh the page. If it does not turn back on you may have done something wrong. Make sure to double check your steps.



**Problems:**

I only had one problem and I needed to troubleshoot the Lambda function. We gave the Lambda function a role allowing it to log into CloudWatch logs which is how I figured out that I copied and pasted the function wrongly. As screenshotted in this documentation at step 3 the python function is wrongly indented on line 15. It should be tabbed twice and it is correct in my GitHub repository so you shouldn’t have any problems.



Make sure the indention on line 15 is correct.