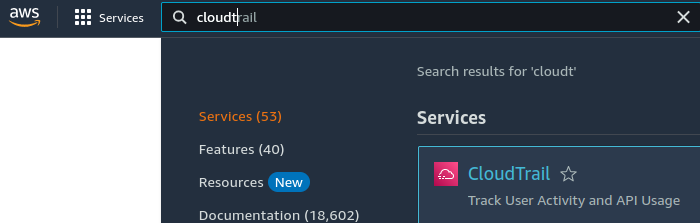
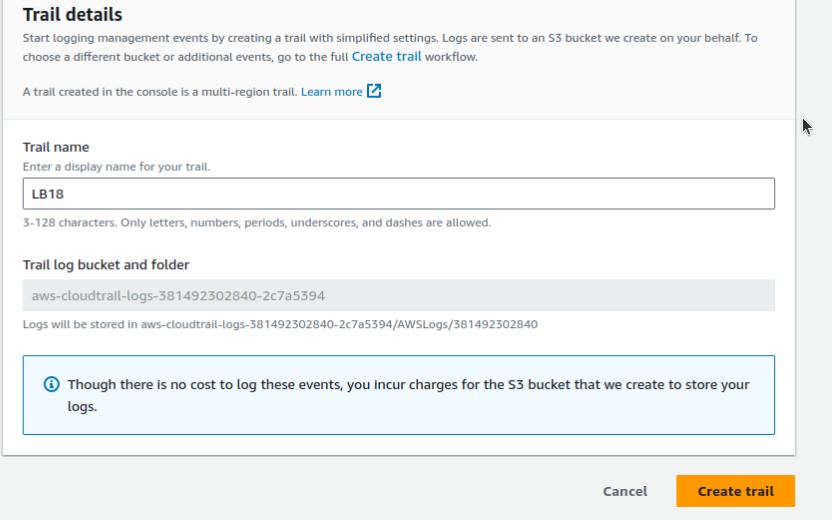
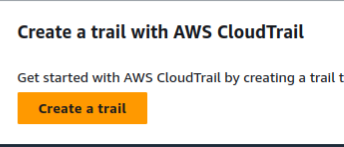
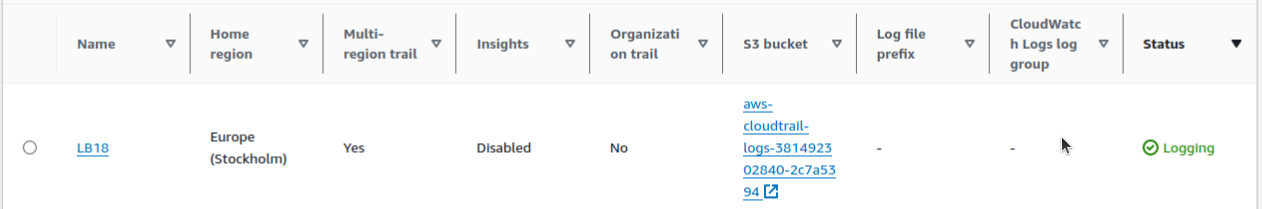
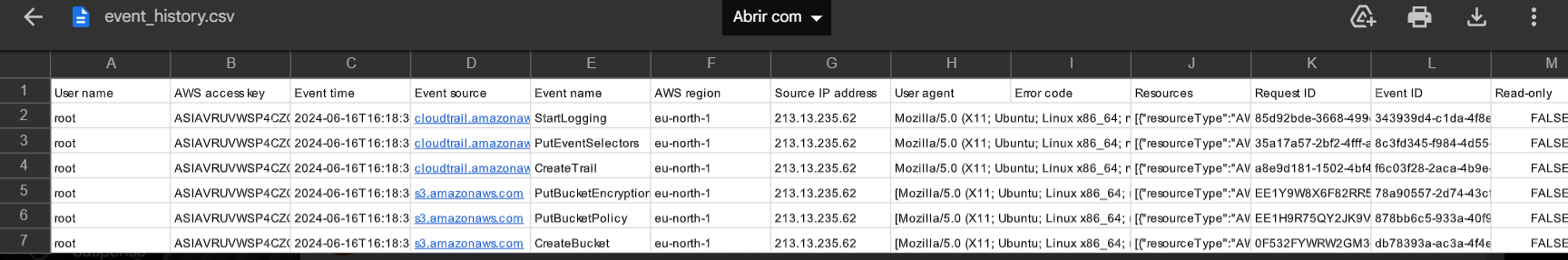
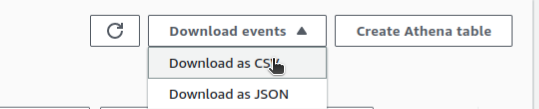
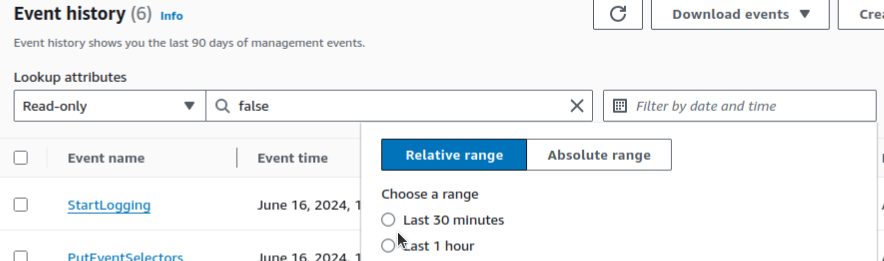
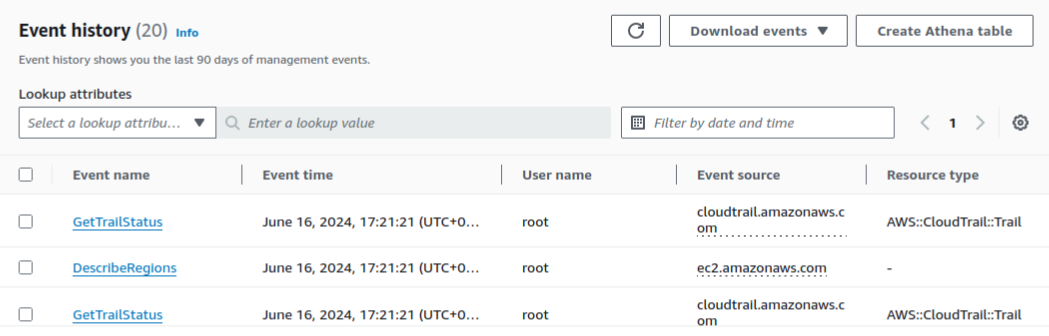
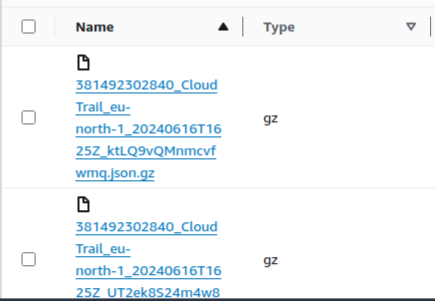
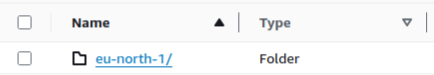
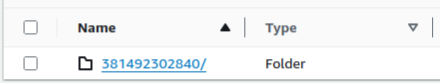
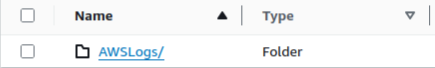
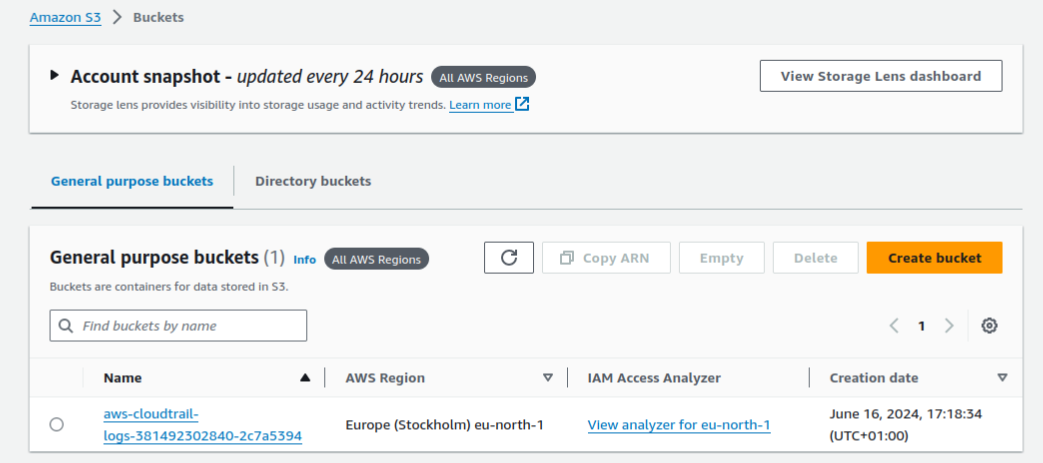
# **Cloud Logging and Monitoring**

### **Part 1: Create a Trail with AWS CloudTrail**

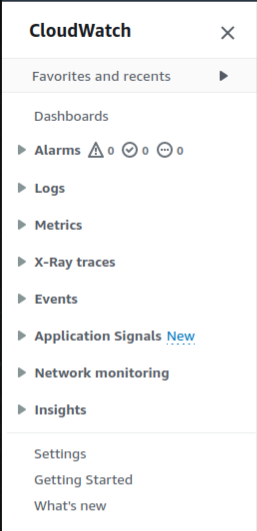
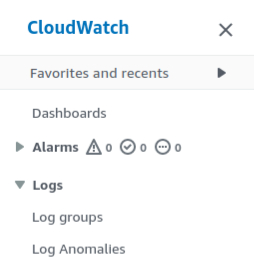
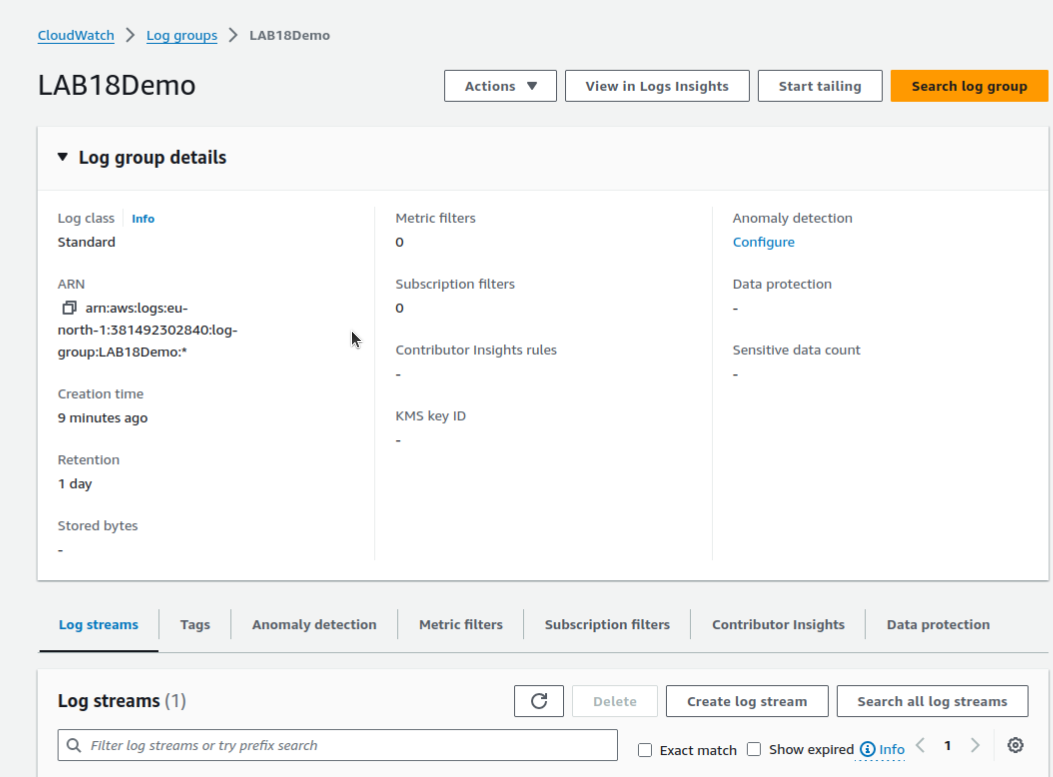
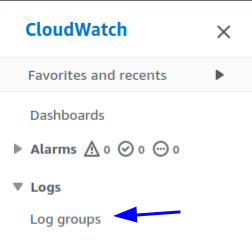
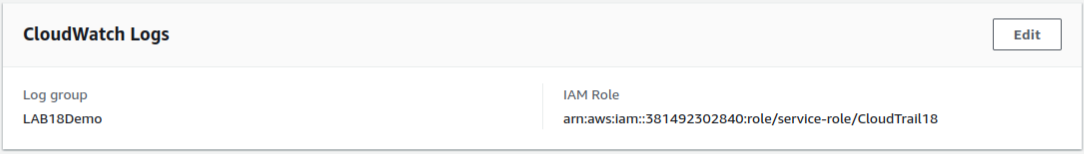
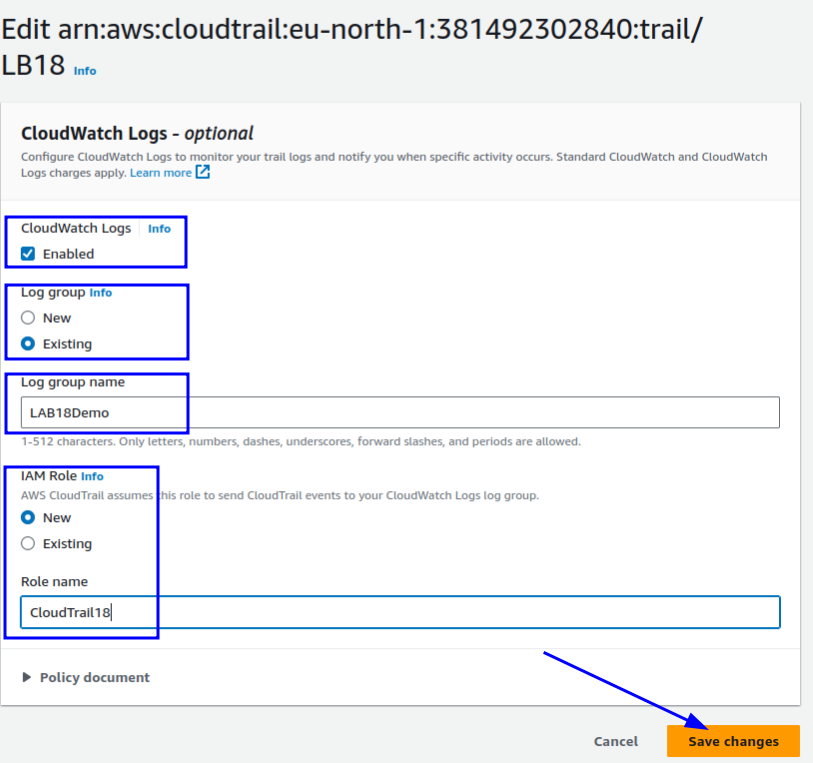
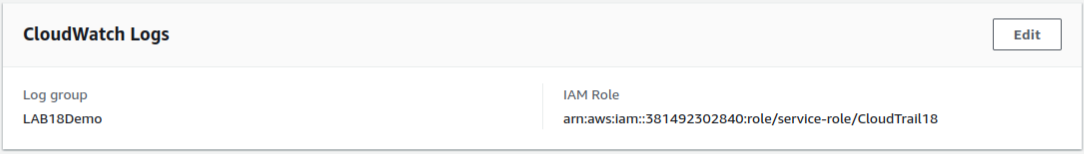
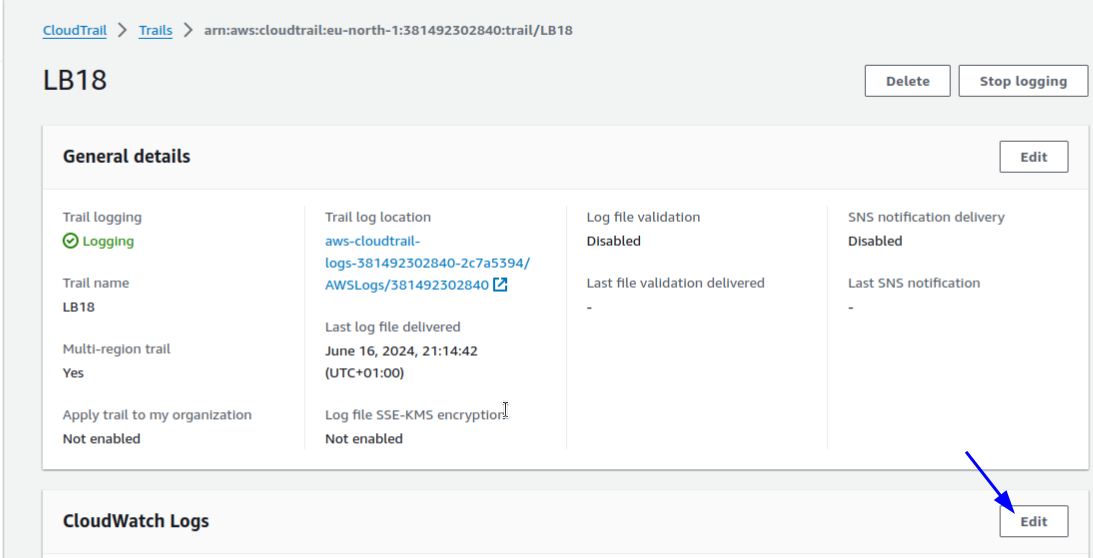
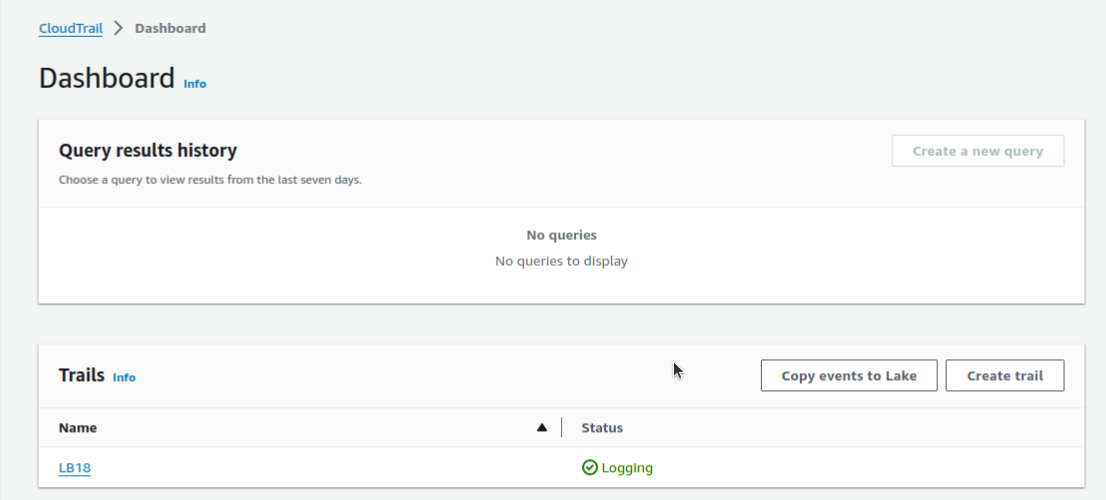
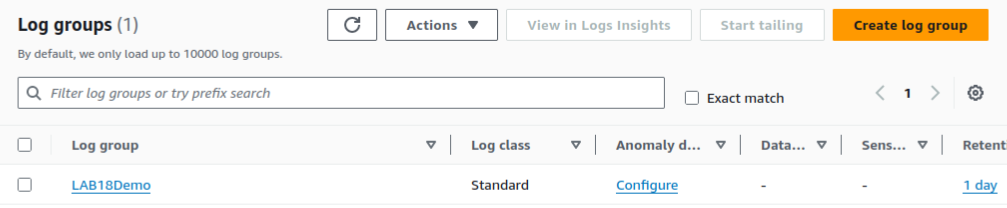
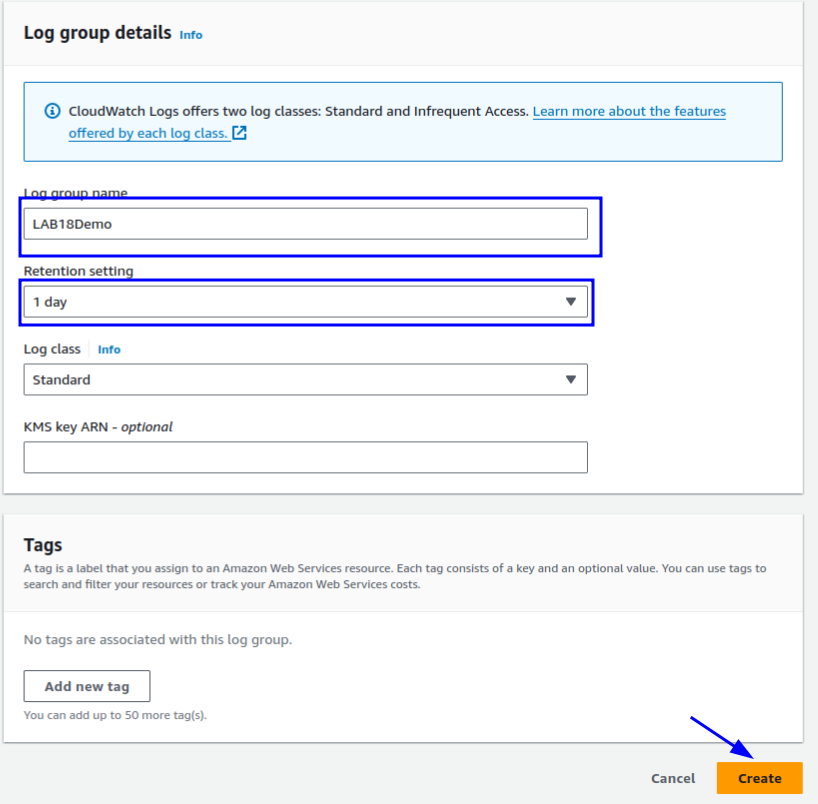
**AWS CloudTrail** is an AWS service that helps you enable governance, compliance, and operational and risk auditing of your AWS account. Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail. Let’s get to know this service by creating our first “trail” object and initialize some logging.

* Access AWS CloudTrail by typing “cloudtrail” into the AWS Management Console search box.
* Using Quick trail create, create a trail for logging management events.
  + Note that an S3 bucket will be created to store your logs. Pay attention to the cost implications of this action.
* Access the trail object and note its state of “Logging”; you can start/stop logging from this screen.
* In CloudTrail console, access event history. Note that CloudTrail records the last 90 days of events for free. Change the time window to something more reasonable like 30m or 1h, then download both a JSON and a CSV of the events to your computer. Take a look at both; which is more useful for post-incident analysis?
* Open the S3 bucket that’s capturing your management logs from Cloudtrail. Drill down into the folders until you can see the logs as .GZ compressed archives. Grab a screenshot of this. It’s always good to know where you’re saving logs.

This is a brief introduction to the capabilities of AWS CloudTrail. Take a look around and get to know this service and its documentation well. Next, let’s make use of this trail.

### **Part 2: Ingest a Trail with Amazon CloudWatch**

Amazon CloudWatch Events can schedule automated actions that trigger when conditions are met. This can be a powerful defensive control to protect you against specific types of activity.

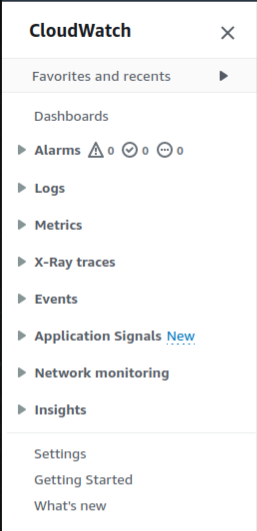
* We’ll need to configure Amazon CloudWatch to ingest log data from CloudTrail, specifically the new Trail you created. Access [this helpful blog](https://aws.amazon.com/blogs/mt/analyzing-cloudtrail-in-cloudwatch/) and complete the sections, “Create a trail in the CloudTrail console” and “Analyzing CloudTrail logs in CloudWatch”. You don’t need to create a new trail, just use the trail you already have.  
    
  
* As a stretch goal, consider completing more activities presented in the blog to gain more expertise in CloudWatch.

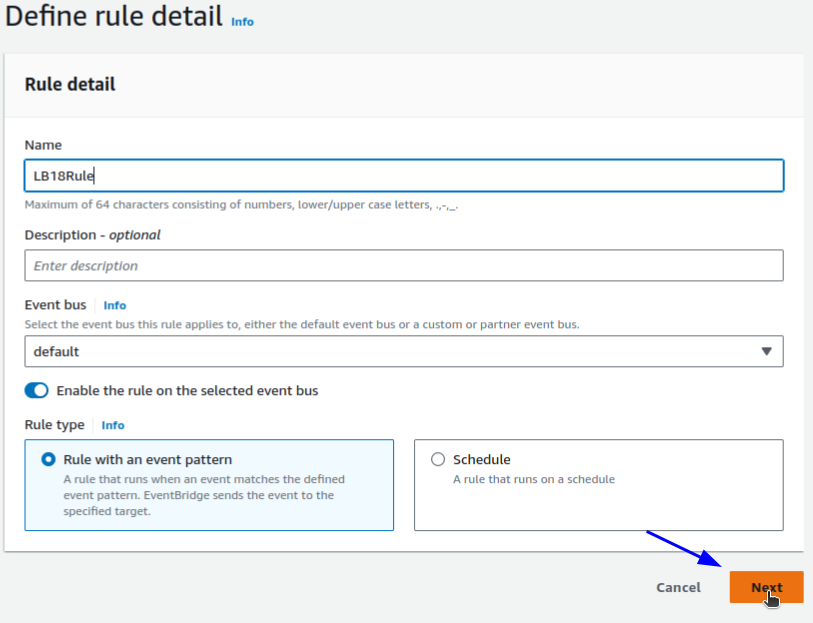
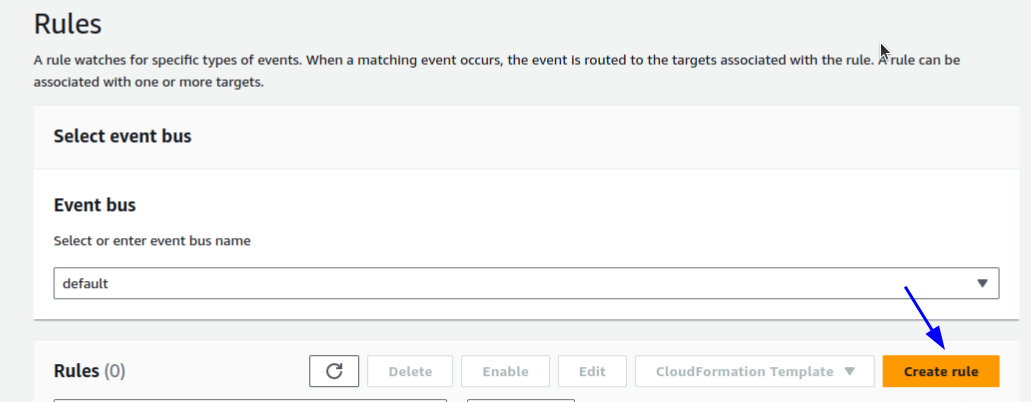
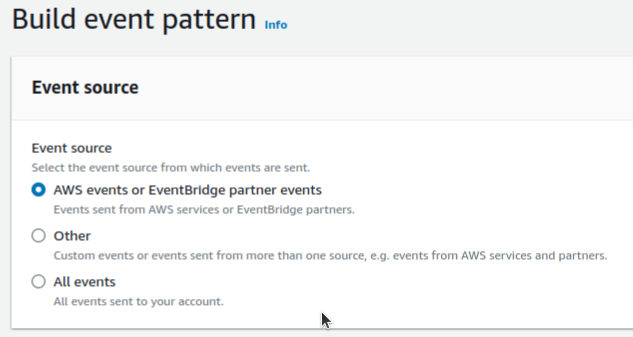
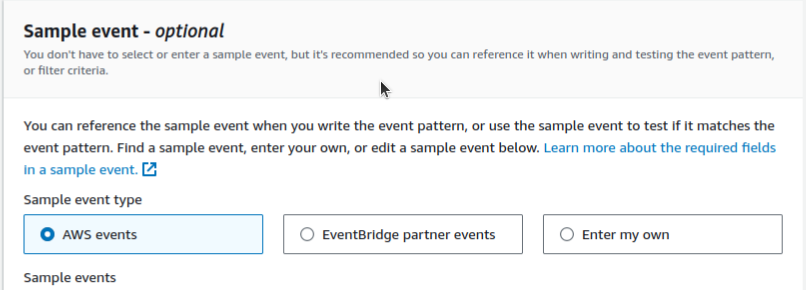
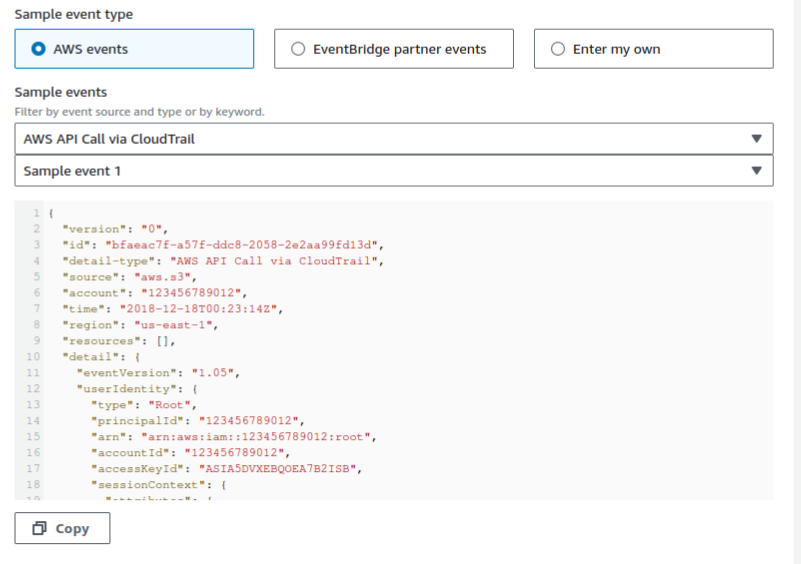
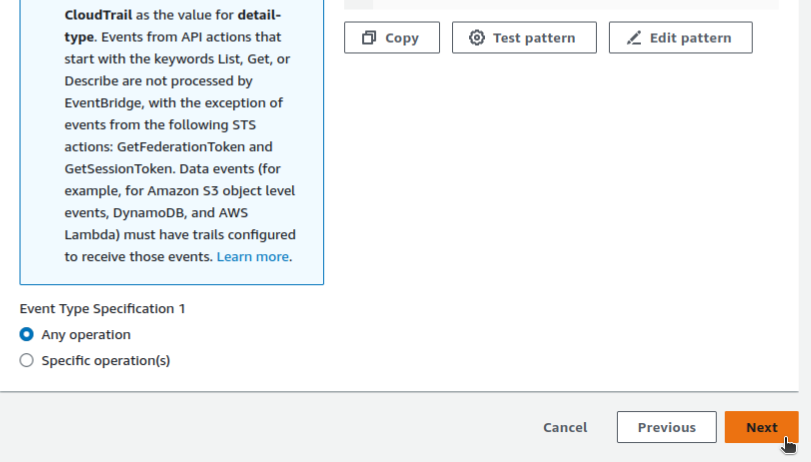
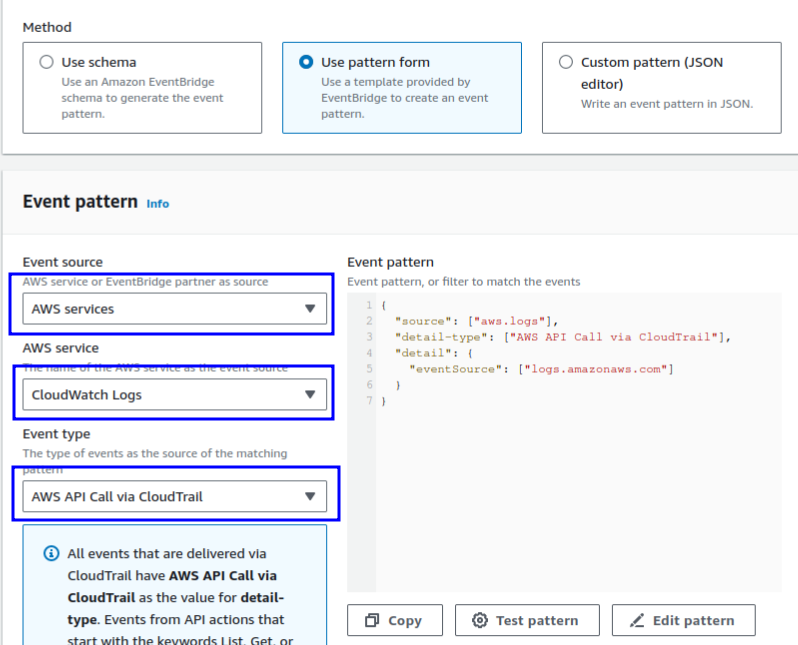
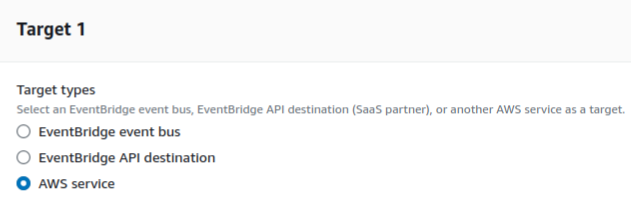
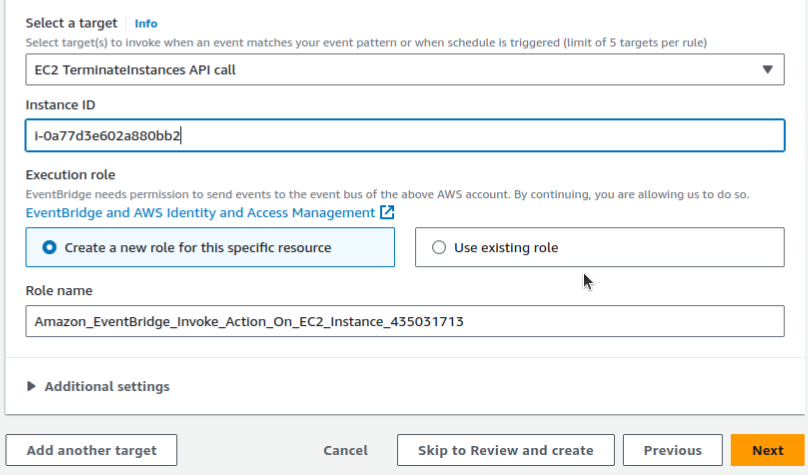
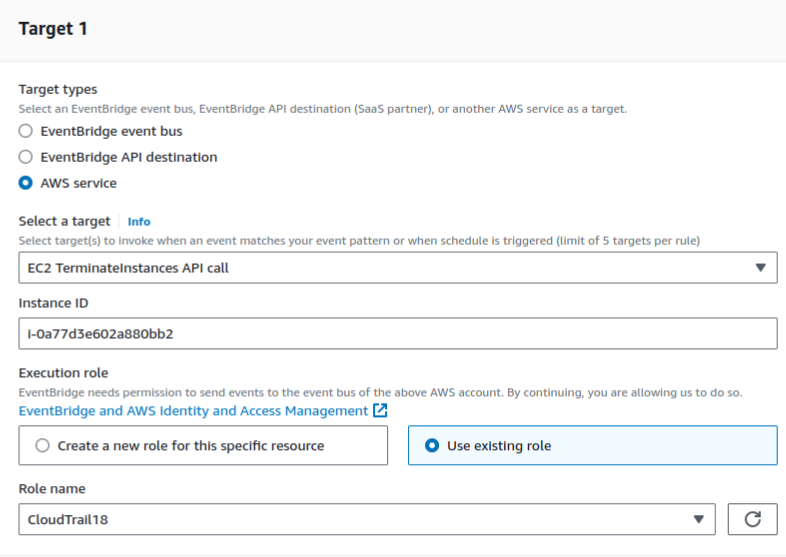
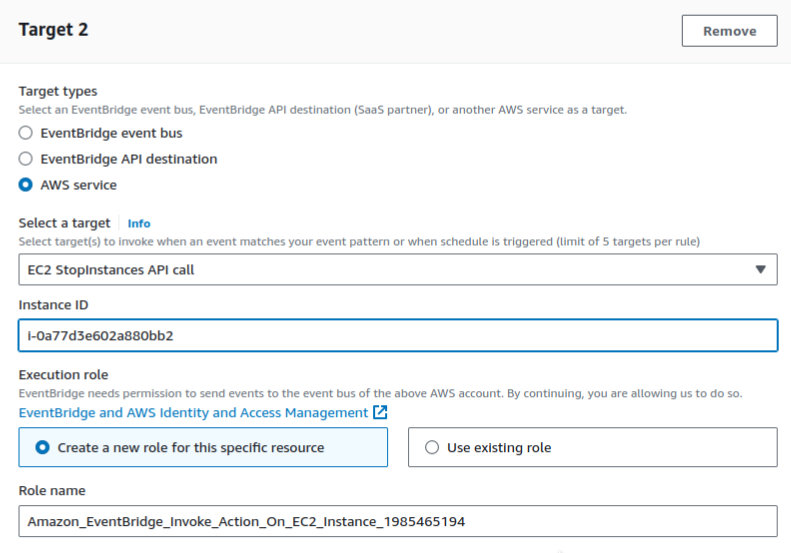
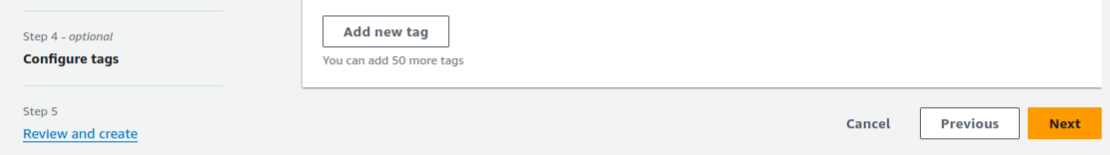
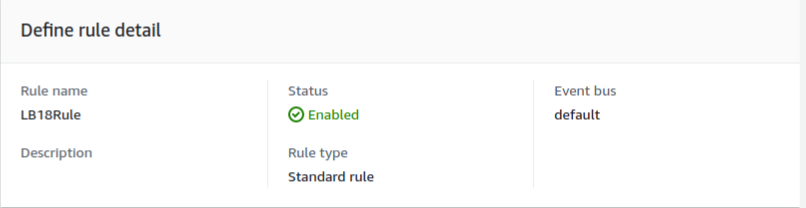
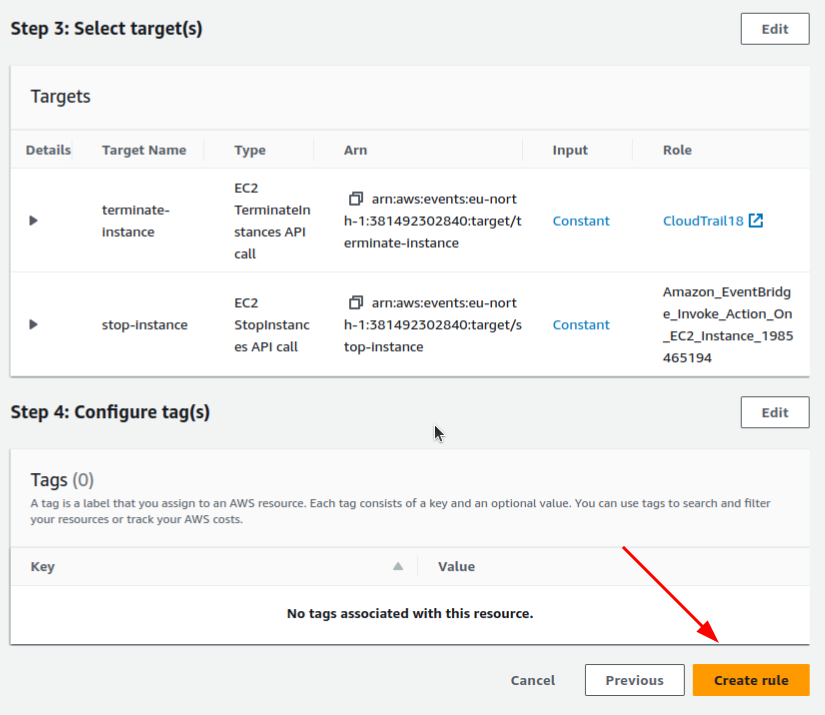
This should give you a rough idea of how CloudTrail feeds data into CloudWatch for analysis.

### **Part 3: Create a CloudWatch Rule**

CloudWatch doesn’t just “watch” the action. It can trigger an “event” that can lead to additional automation in our environment.

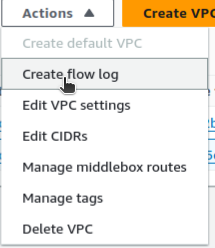
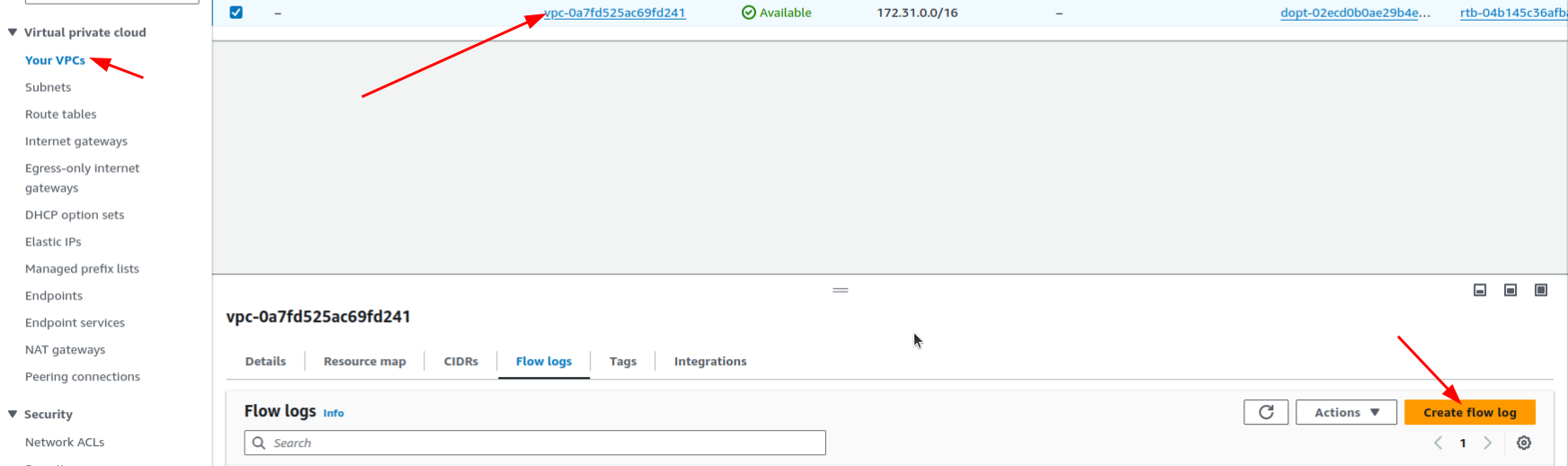
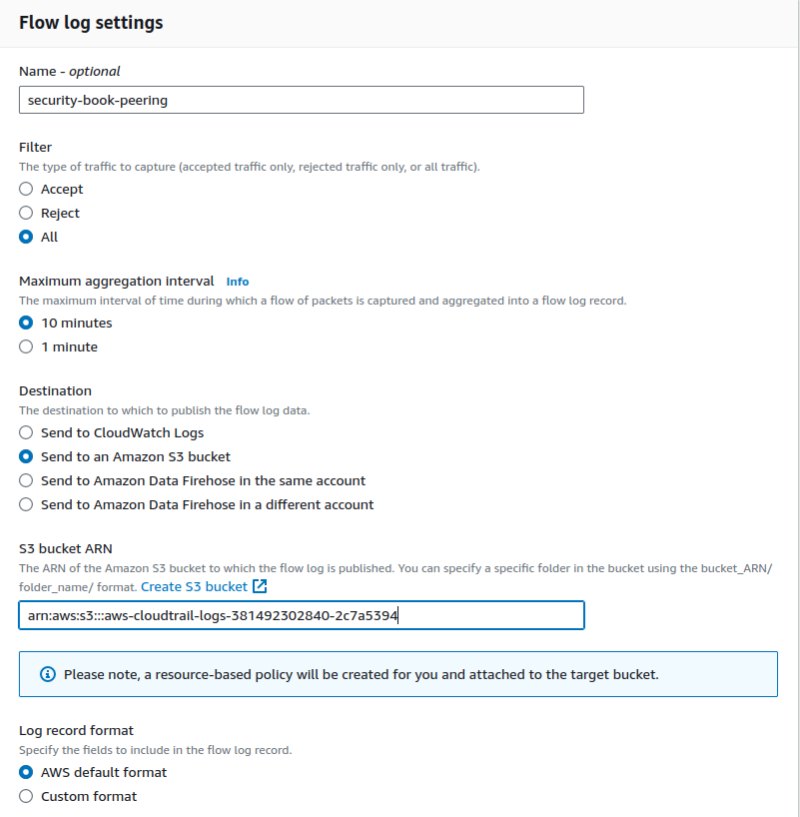
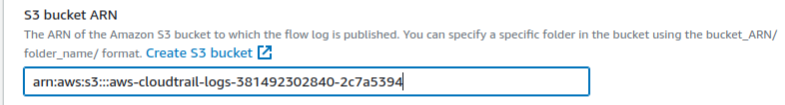
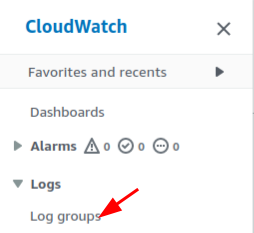
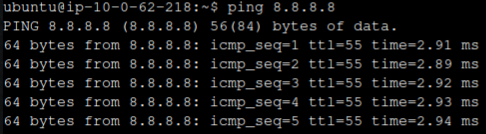
* Open the [CloudWatch console](https://console.aws.amazon.com/cloudwatch/).



* In the navigation pane, choose Events, Create rule.
* For Event source, do the following:
* Choose Event Pattern, Build event pattern to match events by service.
* For Service Name, choose the service that emits the event to trigger the rule.
* For Event Type, choose the specific event that is to trigger the rule. If the only option is AWS API Call via CloudTrail, the selected service does not emit events and you can only base rules on API calls made to this service. For more information about creating this type of rule, see [Creating a CloudWatch Events Rule That Triggers on an AWS API Call Using AWS CloudTrail](https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/Create-CloudWatch-Events-CloudTrail-Rule.html).
* Depending on the service emitting the event, you may see options for Any… and Specific…. Choose Any… to have the event trigger on any type of the selected event, or choose Specific… to choose one or more specific event types.
* For Targets, choose Add Target and choose the AWS service that is to act when an event of the selected type is detected. -In the other fields in this section, enter information specific to this target type, if any is needed.
* For many target types, CloudWatch Events needs permissions to send events to the target. In these cases, CloudWatch Events can create the IAM role needed for your event to run:
* To create an IAM role automatically, choose Create a new role for this specific resource.
* To use an IAM role that you created before, choose Use existing role.
* Optionally, repeat steps 4-6 to add another target for this rule.
* Choose Configure details. For Rule definition, type a name and description for the rule.
* The rule name must be unique within this Region.
* Choose Create rule.

### **Part 4: Create a VPC Flow Log**

**VPC Flow Logs** is a feature that facilitates the capture of IP traffic on VPC network interfaces. We can publish flow log data to Amazon CloudWatch Logs and Amazon S3. In this part of today’s lab, you’ll configure a VPC flow log.

* Under Services > Networking & Content Delivery > VPC > Your VPCs, select your VPC > Flow Logs tab.
* Create a flow log using **security-book-peering** as the trail name. -Select the filter type and maximum aggregation interval.
* Send the flow logs to an S3 bucket by typing **arn:aws:s3:::<YOUR BUCKET NAME>** in the S3 bucket ARN field.
* Test and validate IP traffic is being captured using a flow log.  
    
  {
* "eventVersion": "1.08",
* "userIdentity": {
* "type": "Root",
* "principalId": "381492302840",
* "arn": "arn:aws:iam::381492302840:root",
* "accountId": "381492302840",
* "accessKeyId": "ASIAVRUVWSP4P3ETCGXY",
* "sessionContext": {
* "sessionIssuer": {},
* "webIdFederationData": {},
* "attributes": {
* "creationDate": "2024-06-17T09:10:05Z",
* "mfaAuthenticated": "false"
* }
* },
* "invokedBy": "health.amazonaws.com"
* },
* "eventTime": "2024-06-17T14:25:41Z",
* "eventSource": "health.amazonaws.com",
* "eventName": "DescribeEventAggregates",
* "awsRegion": "us-east-1",
* "sourceIPAddress": "health.amazonaws.com",
* "userAgent": "AWS Internal",
* "requestParameters": {
* "filter": {
* "startTimes": [
* {
* "from": "Jun 10, 2024, 2:25:40 PM"
* }
* ],
* "eventStatusCodes": [
* "open",
* "upcoming"
* ]
* },
* "aggregateField": "eventTypeCategory"
* },
* "responseElements": null,
* "requestID": "df5bfbde-bb2a-45b8-8e4c-273c2c91769e",
* "eventID": "7574b684-8f49-4a48-a304-7679ffe9fe02",
* "readOnly": true,
* "eventType": "AwsApiCall",
* "managementEvent": true,
* "recipientAccountId": "381492302840",
* "eventCategory": "Management",
* "sessionCredentialFromConsole": "true"
* }