Dhruval Rana

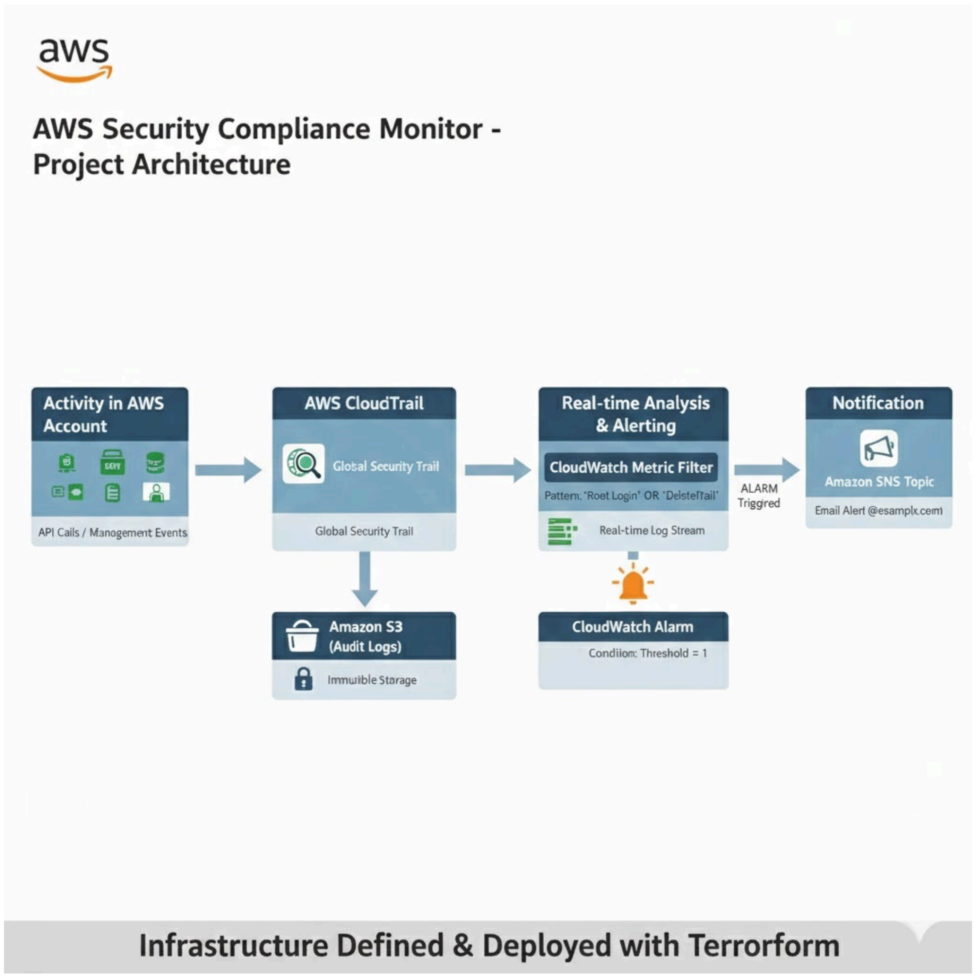
AWS Security Compliance Monitor

Project overview:

- **The Goal:** Automatically monitor AWS accounts for "Management Events" (like deleting a database or stopping a trail) and send a real-time alert to email.
- **The Security Value:** Creates an **Audit Trail** and **Real-time Alerting** system, which are the first two steps in any professional security framework (like CIS or NIST).
- **Automated Deployment:** Used Terraform to provision the **S3 Bucket** for log storage, the **CloudTrail** for recording actions, and the **CloudWatch Alarms** that act as security tripwires.

Infrastructure Architecture:

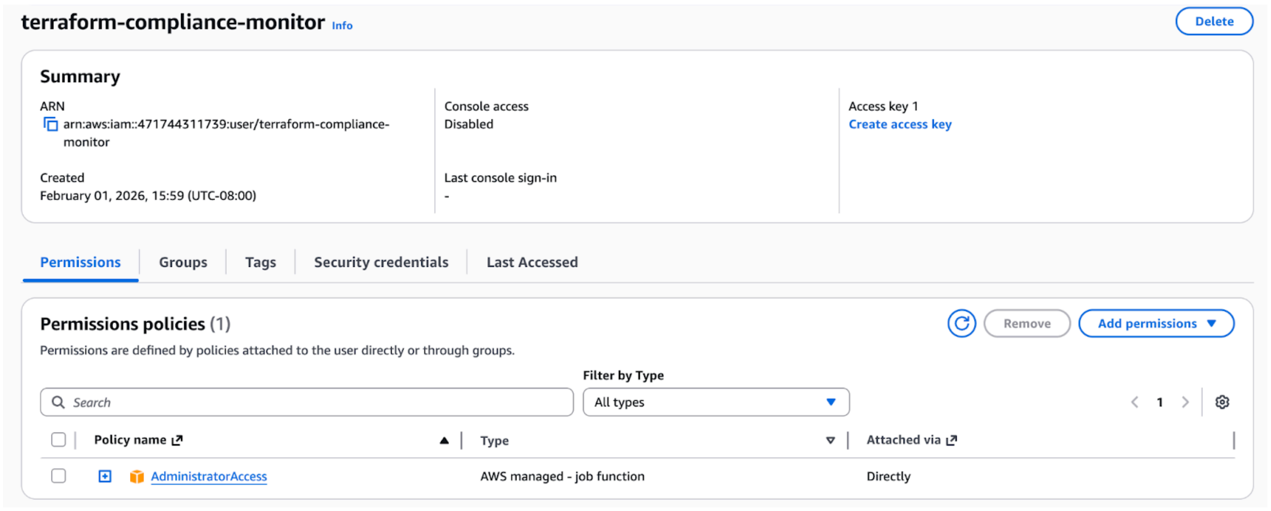
- **CloudTrail:** Tracks every API call made in account.
- **CloudWatch Logs:** Receives the data from CloudTrail.
- **Metric Filters:** Scans the logs for specific "bad" actions (e.g., `DeleteTrail`).
- **CloudWatch Alarm:** Triggers when the filter finds a match.
- **SNS (Simple Notification Service):** Sends an email



Dhruval Rana

Step 1: Created an IAM User

- **Accessed IAM:** Logged into the AWS Management Console and navigated to the **Identity and Access Management (IAM)** service to manage security identities.
- **Created a New User:** Initiated the creation process by clicking **Users** and then **Create user** to avoid using the root account for daily tasks.
- **Defined the Username:** Set the user identity name as `terraform-compliance-monior` to clearly distinguish it as a programmatic account.
- **Set Permission Strategy:** Selected **Attach policies directly** to ensure specific permissions were linked directly to this user.
- **Granted Administrator Access:** Searched for and selected the `AdministratorAccess` policy, providing full permissions to manage AWS resources for this learning project.



terraform-compliance-monitor Info

Summary

ARN
arn:aws:iam::471744311739:user/terraform-compliance-monitor

Created
February 01, 2026, 15:59 (UTC-08:00)

Console access
Disabled

Last console sign-in
-

Access key 1
Create access key

Permissions policies (1)

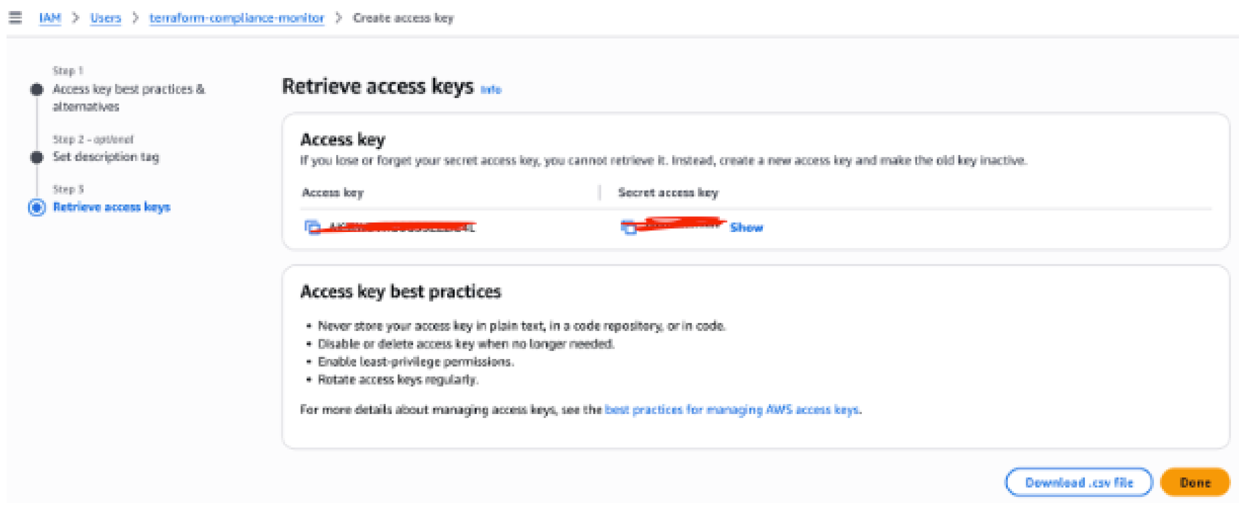
Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly

Step 2: Generated Access Keys

- **Navigated to User Security Credentials:** Went to the specific user profile (in this case, `terraform-compliance-monitor`) within the IAM dashboard.
- **Initiated Access Key Creation:** Clicked on the Security credentials tab and selected Create access key.
- **Selected Use Case:** Selected Command Line Interface (CLI) as the use case to enable Terraform to communicate with AWS from a local machine.
- **Set Description Tag:** Moved through the optional step of adding a description tag to identify what these keys are used for.
- **Generated the Credentials:** Reached the final "Retrieve access keys" screen which generated Access key ID and Secret access key.
- **Secured the Keys:** Presented with the option to Download .csv file, which is the only way to save the Secret Access Key before it becomes permanently hidden.



IAM > Users > terraform-compliance-monitor > Create access key

Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
[redacted]	[redacted] Show

Access key best practices

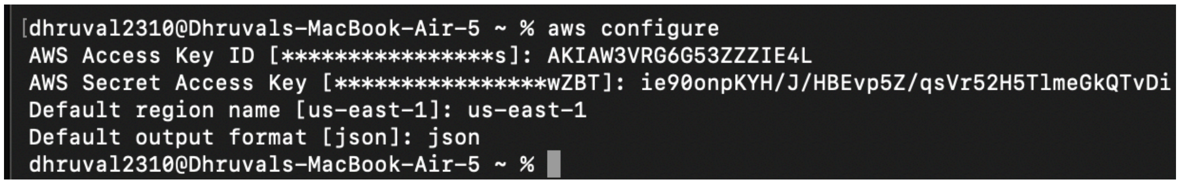
- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

Download .csv file   Done

Step 3: Linked local environment to AWS account

- **Launched Terminal:** Opened command-line interface, such as Command Prompt, PowerShell, or Terminal.
- **Initiated Configuration:** Typed the command "aws configure" to start the setup process.
- **Entered Identity ID:** Prompted for AWS Access Key ID, pasted the long alphanumeric string (starting with AKIA) from previous step.
- **Entered Secret Key:** For the AWS Secret Access Key, pasted the private string that was saved or downloaded.
- **Set Geographic Region:** Specified us-east-1 as the Default region name to tell AWS where to create resources.
- **Choose Data Format:** Entered json for the Default output format to ensure the terminal returns information in a standard, readable structure.

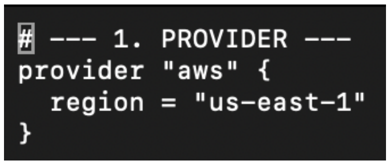


```
[dhruval2310@Dhruvals-MacBook-Air-5 ~ % aws configure
 AWS Access Key ID [****************s]: AKIAW3VRG6G53ZZZIE4L
 AWS Secret Access Key [****************wZBT]: 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
 Default region name [us-east-1]: us-east-1
 Default output format [json]: json
 dhruval2310@Dhruvals-MacBook-Air-5 ~ %
```

Step 4: Terraform script in CLI

1. The Provider Block

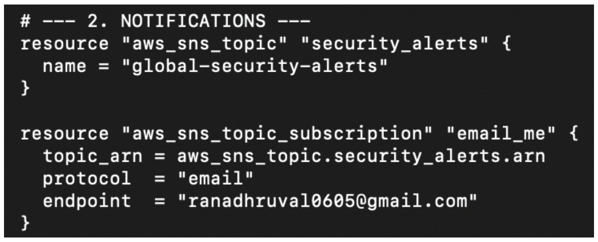
- **Purpose:** This tells Terraform which cloud platform is using.
- **Action:** It initializes the **AWS** provider and targets the **us-east-1** region for all resource deployments.



```
# --- 1. PROVIDER ---
provider "aws" {
  region = "us-east-1"
}
```

2. The Notifications Block

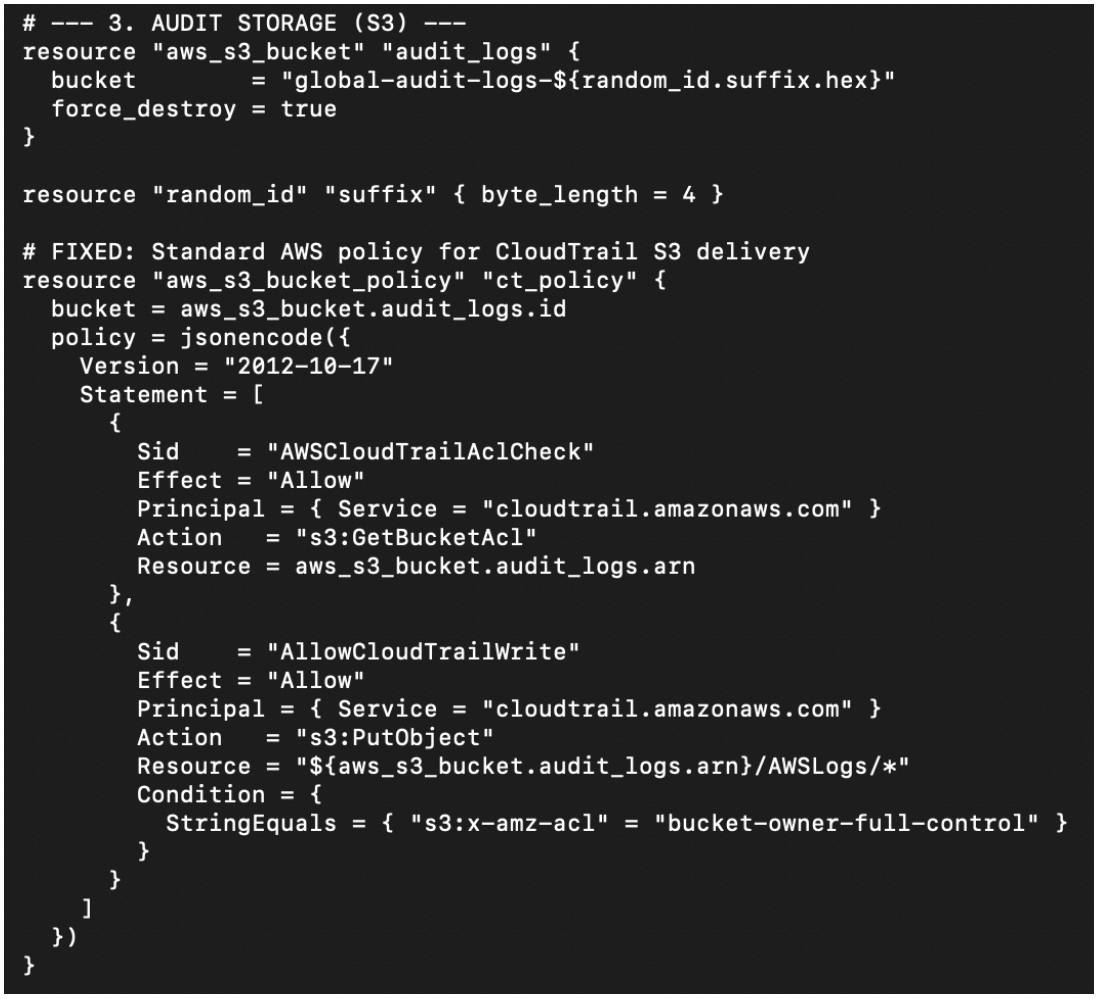
- **SNS Topic:** Creates a communication hub called global-security-alerts.
- **Subscription:** Connects specific email (**ranadhruval0605@gmail.com**) to that hub so receive the actual alerts.



```
# --- 2. NOTIFICATIONS ---
resource "aws_sns_topic" "security_alerts" {
  name = "global-security-alerts"
}

resource "aws_sns_topic_subscription" "email_me" {
  topic_arn = aws_sns_topic.security_alerts.arn
  protocol  = "email"
  endpoint  = "ranadhruval0605@gmail.com"
}
```

3. Audit Storage (S3) Block

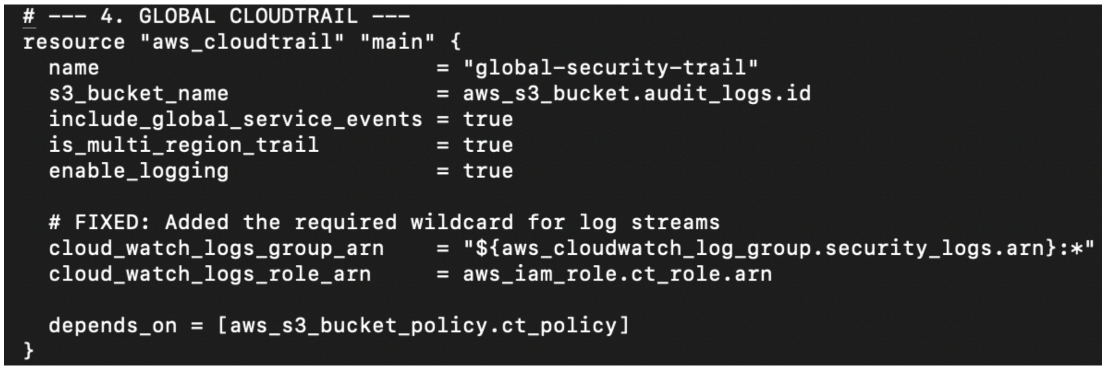
- **Bucket Creation:** Sets up an **S3 bucket** to act as a permanent, "cold storage" vault for every action performed in account.
- **Random Suffix:** Uses a **random ID generator** to ensure the bucket name is globally unique, which is a requirement for AWS S3.
- **Bucket Policy:** This is a "lock and key" instruction that specifically gives the **CloudTrail service** permission to write files into bucket while keeping everyone else out.



```
# --- 3. AUDIT STORAGE (S3) ---
resource "aws_s3_bucket" "audit_logs" {
  bucket        = "global-audit-logs-${random_id.suffix.hex}"
  force_destroy = true
}

resource "random_id" "suffix" { byte_length = 4 }

# FIXED: Standard AWS policy for CloudTrail S3 delivery
resource "aws_s3_bucket_policy" "ct_policy" {
  bucket = aws_s3_bucket.audit_logs.id
  policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Sid    = "AWSCloudTrailAclCheck"
        Effect = "Allow"
        Principal = { Service = "cloudtrail.amazonaws.com" }
        Action   = "s3:GetBucketAcl"
        Resource = aws_s3_bucket.audit_logs.arn
      },
      {
        Sid    = "AllowCloudTrailWrite"
        Effect = "Allow"
        Principal = { Service = "cloudtrail.amazonaws.com" }
        Action   = "s3:PutObject"
        Resource = "${aws_s3_bucket.audit_logs.arn}/AWSLogs/*"
        Condition = {
          StringEquals = { "s3:x-amz-acl" = "bucket-owner-full-control" }
        }
      }
    ]
  })
}
```

4. Global CloudTrail Block

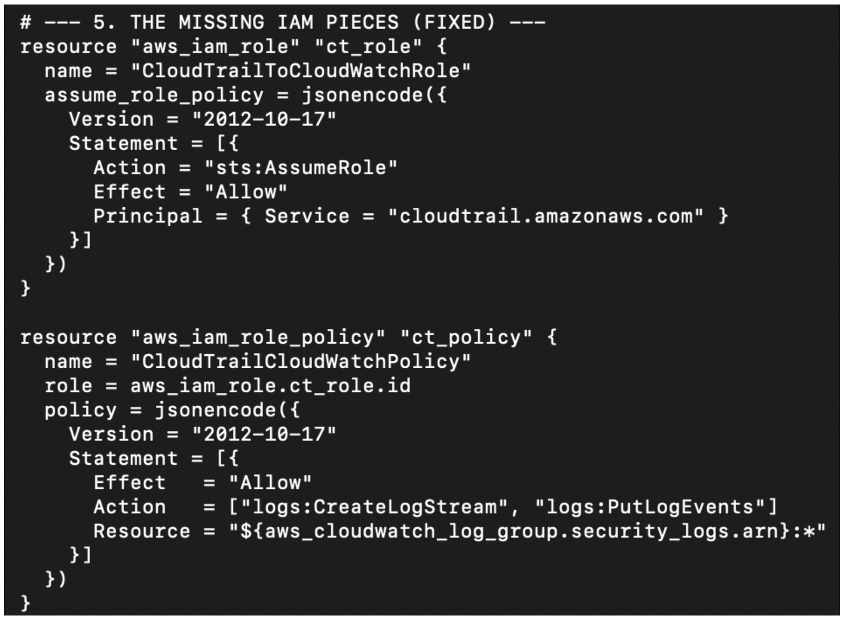
- **The Recorder:** Activates **CloudTrail**, which functions like a security camera recording every API call (login, resource deletion, etc.).
- **Global Scope:** It is configured as a **Multi-Region trail**, meaning it watches all AWS regions simultaneously, not just default one.
- **CloudWatch Integration:** Links the trail to **CloudWatch Logs** so the recordings are sent to a live stream for immediate analysis rather than just sitting in a storage bucket.



```
# --- 4. GLOBAL CLOUDTRAIL ---
resource "aws_cloudtrail" "main" {
  name                          = "global-security-trail"
  s3_bucket_name                = aws_s3_bucket.audit_logs.id
  include_global_service_events = true
  is_multi_region_trail         = true
  enable_logging                = true

  # FIXED: Added the required wildcard for log streams
  cloud_watch_logs_group_arn    = "${aws_cloudwatch_log_group.security_logs.arn}:*"
  cloud_watch_logs_role_arn     = aws_iam_role.ct_role.arn

  depends_on = [aws_s3_bucket_policy.ct_policy]
}
```

5. IAM Permissions Block

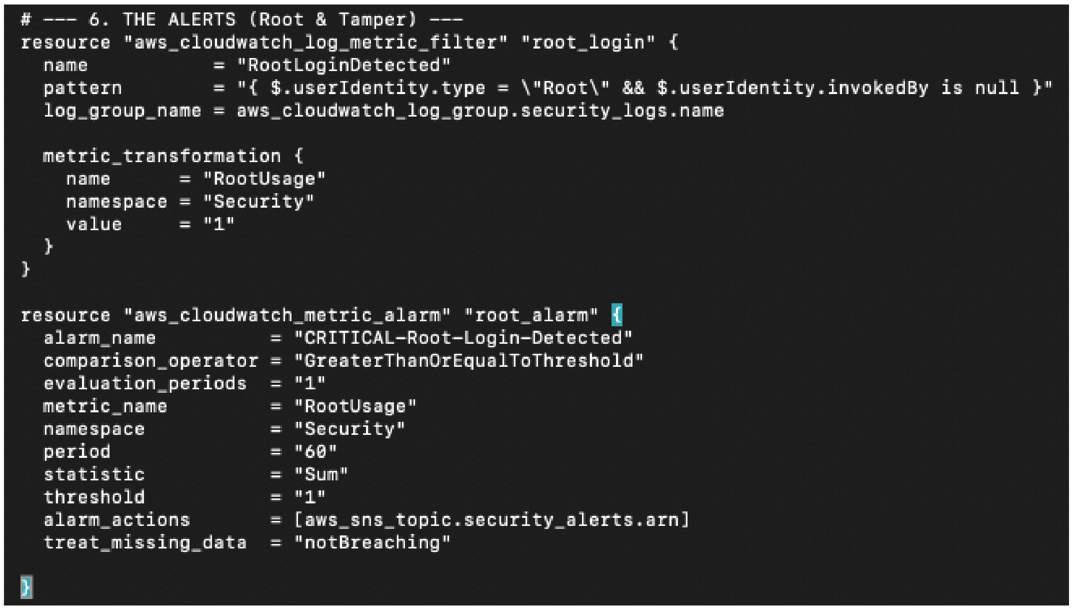
- **Role Creation:** Creates a "trusted identity" (CloudTrailToCloudWatchRole) that the CloudTrail service can "wear" to perform tasks.
- **Policy Attachment:** Attaches specific permissions to that role, allowing it to **CreateLogStream** and **PutLogEvents** inside CloudWatch logs.



```
# --- 5. THE MISSING IAM PIECES (FIXED) ---
resource "aws_iam_role" "ct_role" {
  name = "CloudTrailToCloudWatchRole"
  assume_role_policy = jsonencode({
    Version = "2012-10-17"
    Statement = [{
      Action = "sts:AssumeRole"
      Effect = "Allow"
      Principal = { Service = "cloudtrail.amazonaws.com" }
    }]
  })
}

resource "aws_iam_role_policy" "ct_policy" {
  name = "CloudTrailCloudWatchPolicy"
  role = aws_iam_role.ct_role.id
  policy = jsonencode({
    Version = "2012-10-17"
    Statement = [{
      Effect   = "Allow"
      Action   = ["logs:CreateLogStream", "logs:PutLogEvents"]
      Resource = "${aws_cloudwatch_log_group.security_logs.arn}:*"
    }]
  })
}
```

6. The Alerts (Root & Tamper) Block

- **Metric Filter:** This acts as a "search filter" that constantly scans incoming logs for the specific pattern of a **Root User login**.
- **Metric Transformation:** Every time the filter finds a match, it increments a counter (a metric) by **1**.
- **CloudWatch Alarm:** This is the "tripwire." If the counter hits **1 or more**, it triggers analarm that sends a message to **SNS Topic**, which then emails immediately.



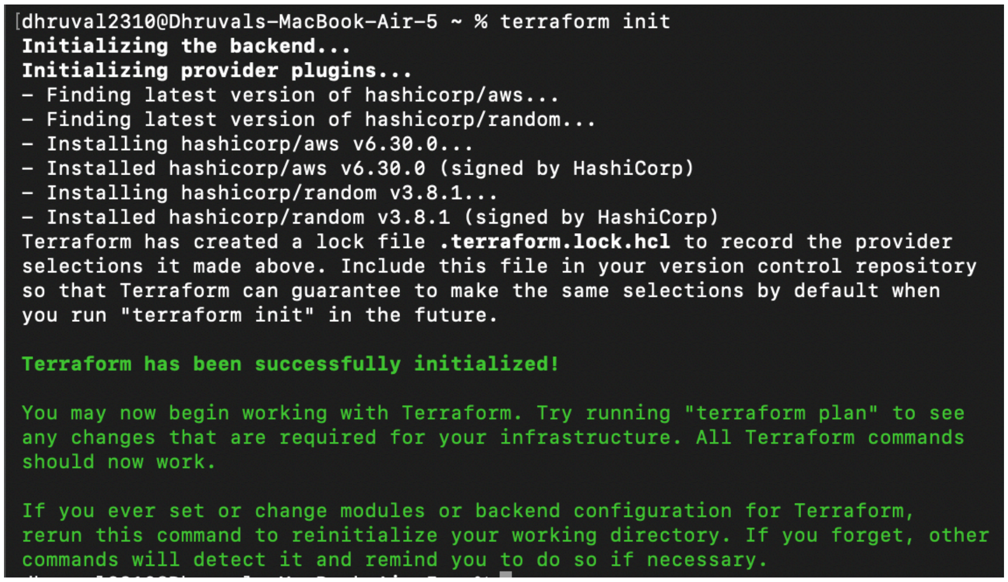
```
# --- 6. THE ALERTS (Root & Tamper) ---
resource "aws_cloudwatch_log_metric_filter" "root_login" {
  name           = "RootLoginDetected"
  pattern        = "{ $.userIdentity.type = \"Root\" && $.userIdentity.invokedBy is null }"
  log_group_name = aws_cloudwatch_log_group.security_logs.name

  metric_transformation {
    name      = "RootUsage"
    namespace = "Security"
    value     = "1"
  }
}

resource "aws_cloudwatch_metric_alarm" "root_alarm" {
  alarm_name          = "CRITICAL-Root-Login-Detected"
  comparison_operator = "GreaterThanOrEqualToThreshold"
  evaluation_periods  = "1"
  metric_name         = "RootUsage"
  namespace           = "Security"
  period              = "60"
  statistic           = "Sum"
  threshold           = "1"
  alarm_actions       = [aws_sns_topic.security_alerts.arn]
  treat_missing_data  = "notBreaching"

}
```

Dhruval Rana

Step 5: Tested the Connection

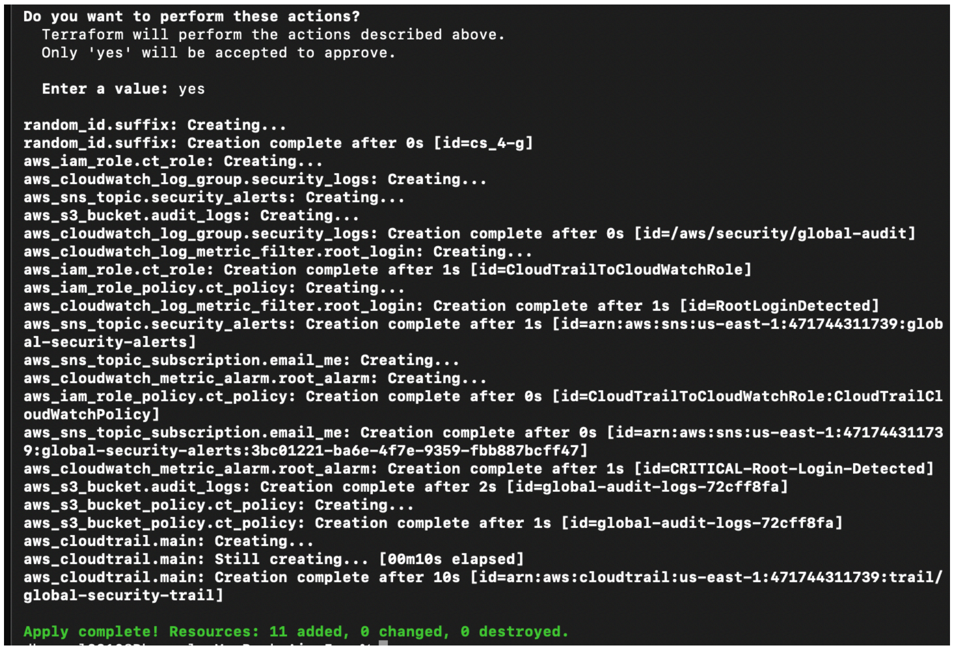
To finalize setup and begin deployment

- **Initialized the Project:** Ran `terraform init`, which tells Terraform to scan code and download the necessary **AWS provider** plugins so it knows how to communicate with the AWS API.
- **Verified Permissions:** Executed `terraform plan`, which acts as a "dry run" to verify that `terraform-user` keys are correctly configured and have the permissions needed to read AWS environment.
- **Confirmed Resource Changes:** During the plan phase, reviewed the terminal output to see exactly which resources (S3, CloudTrail, Alarms) Terraform intended to create.



```
[dhruval2310@Dhruvals-MacBook-Air-5 ~ % terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/random...
- Installing hashicorp/aws v6.30.0...
- Installed hashicorp/aws v6.30.0 (signed by HashiCorp)
- Installing hashicorp/random v3.8.1...
- Installed hashicorp/random v3.8.1 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

- **Launched Deployment:** Ran `terraform apply`, the final command that sends instructions to AWS to start building the security infrastructure.
- **Authorized the Build:** Typed `yes` when prompted by the terminal to confirm wanted to proceed with the actual resource creation in account.

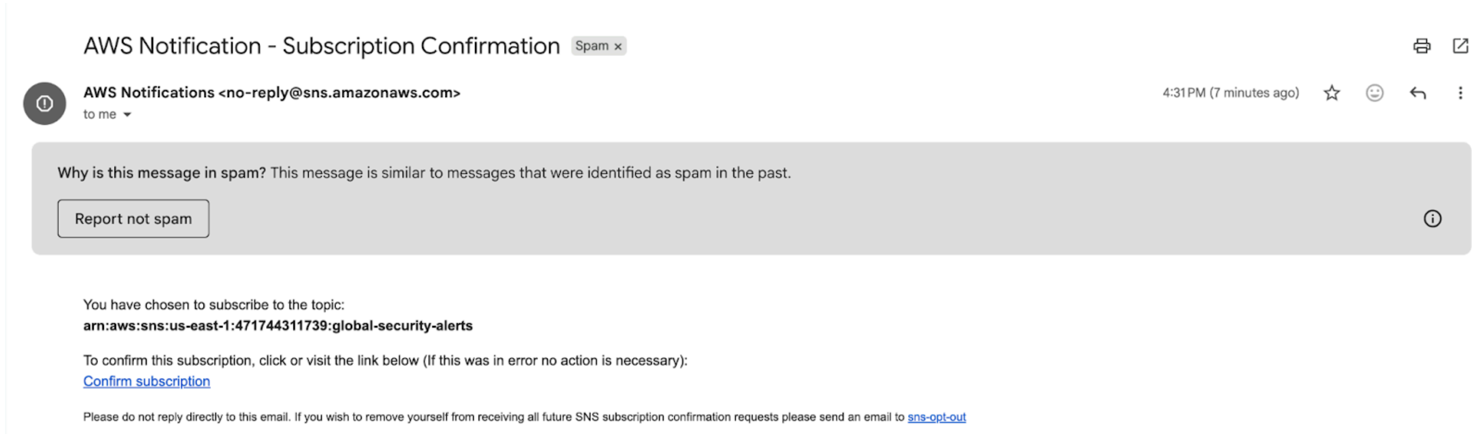


```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

random_id.suffix: Creating...
random_id.suffix: Creation complete after 0s [id=cs_4-g]
aws_iam_role.ct_role: Creating...
aws_cloudwatch_log_group.security_logs: Creating...
aws_sns_topic.security_alerts: Creating...
aws_s3_bucket.audit_logs: Creating...
aws_cloudwatch_log_group.security_logs: Creation complete after 0s [id=/aws/security/global-audit]
aws_cloudwatch_log_metric_filter.root_login: Creating...
aws_iam_role.ct_role: Creation complete after 1s [id=CloudTrailToCloudWatchRole]
aws_iam_role_policy.ct_policy: Creating...
aws_cloudwatch_log_metric_filter.root_login: Creation complete after 1s [id=RootLoginDetected]
aws_sns_topic.security_alerts: Creation complete after 1s [id=arn:aws:sns:us-east-1:471744311739:glob
al-security-alerts]
aws_sns_topic_subscription.email_me: Creating...
aws_cloudwatch_metric_alarm.root_alarm: Creating...
aws_iam_role_policy.ct_policy: Creation complete after 0s [id=CloudTrailToCloudWatchRole:CloudTrailCl
oudWatchPolicy]
aws_sns_topic_subscription.email_me: Creation complete after 0s [id=arn:aws:sns:us-east-1:47174431173
9:global-security-alerts:3bc01221-ba6e-4f7e-9359-fbb887bcff47]
aws_cloudwatch_metric_alarm.root_alarm: Creation complete after 1s [id=CRITICAL-Root-Login-Detected]
aws_s3_bucket.audit_logs: Creation complete after 2s [id=global-audit-logs-72cff8fa]
aws_s3_bucket_policy.ct_policy: Creating...
aws_s3_bucket_policy.ct_policy: Creation complete after 1s [id=global-audit-logs-72cff8fa]
aws_cloudtrail.main: Creating...
aws_cloudtrail.main: Still creating... [00m10s elapsed]
aws_cloudtrail.main: Creation complete after 10s [id=arn:aws:cloudtrail:us-east-1:471744311739:trail/
global-security-trail]

Apply complete! Resources: 11 added, 0 changed, 0 destroyed.
```

Step 6: Activated the Alert Pipeline

Even though Terraform built the infrastructure, AWS requires a manual "opt-in" to prevent spam.

- **Monitored Inbox:** Checked email for a message from AWS Notifications with the subject "Subscription Confirmation."
- **Located the Request:** Opened the email which identified the specific SNS topic created (`global-security-alerts`).
- **Confirmed Subscription:** Clicked the Confirm subscription link within the email.
- **Verified Activation:** Reached the "Subscription confirmed!" landing page, which officially authorized AWS to send security alerts to email address.



AWS Notification - Subscription Confirmation Spam ×

AWS Notifications <no-reply@sns.amazonaws.com>
to me

4:31 PM (7 minutes ago)

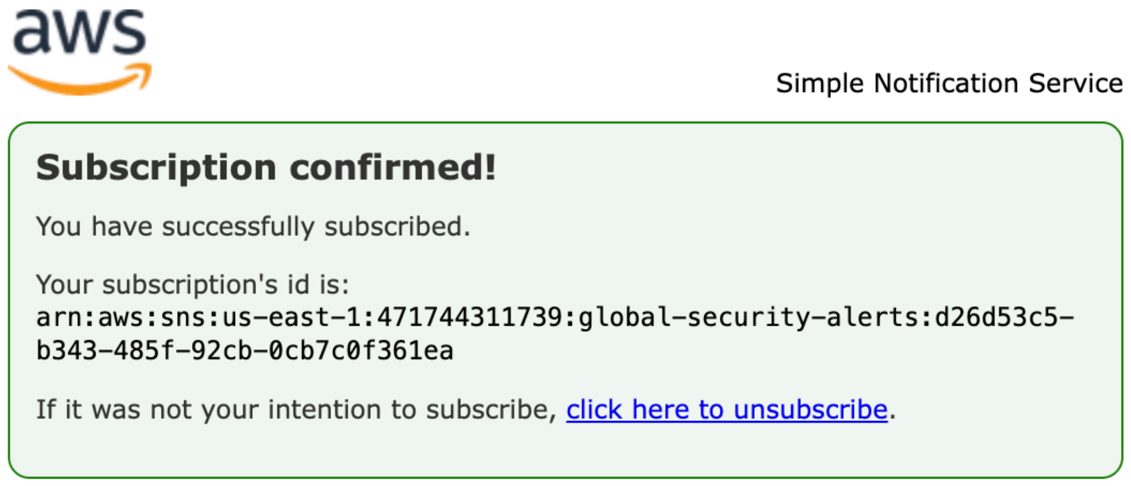
Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Report not spam

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:471744311739:global-security-alerts

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out



Simple Notification Service

Subscription confirmed!

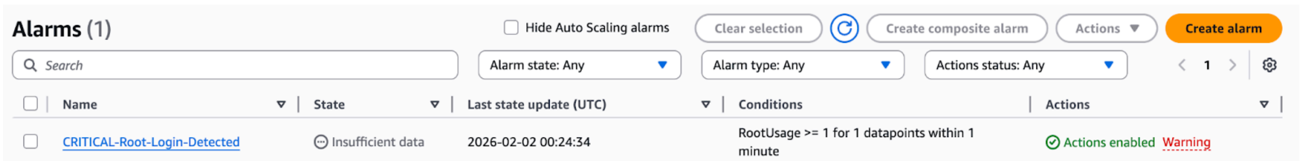
You have successfully subscribed.

Your subscription's id is:
arn:aws:sns:us-east-1:471744311739:global-security-alerts:d26d53c5-b343-485f-92cb-0cb7c0f361ea

If it was not your intention to subscribe, click here to unsubscribe.

Step 7: Verified for CloudWatch Alarm

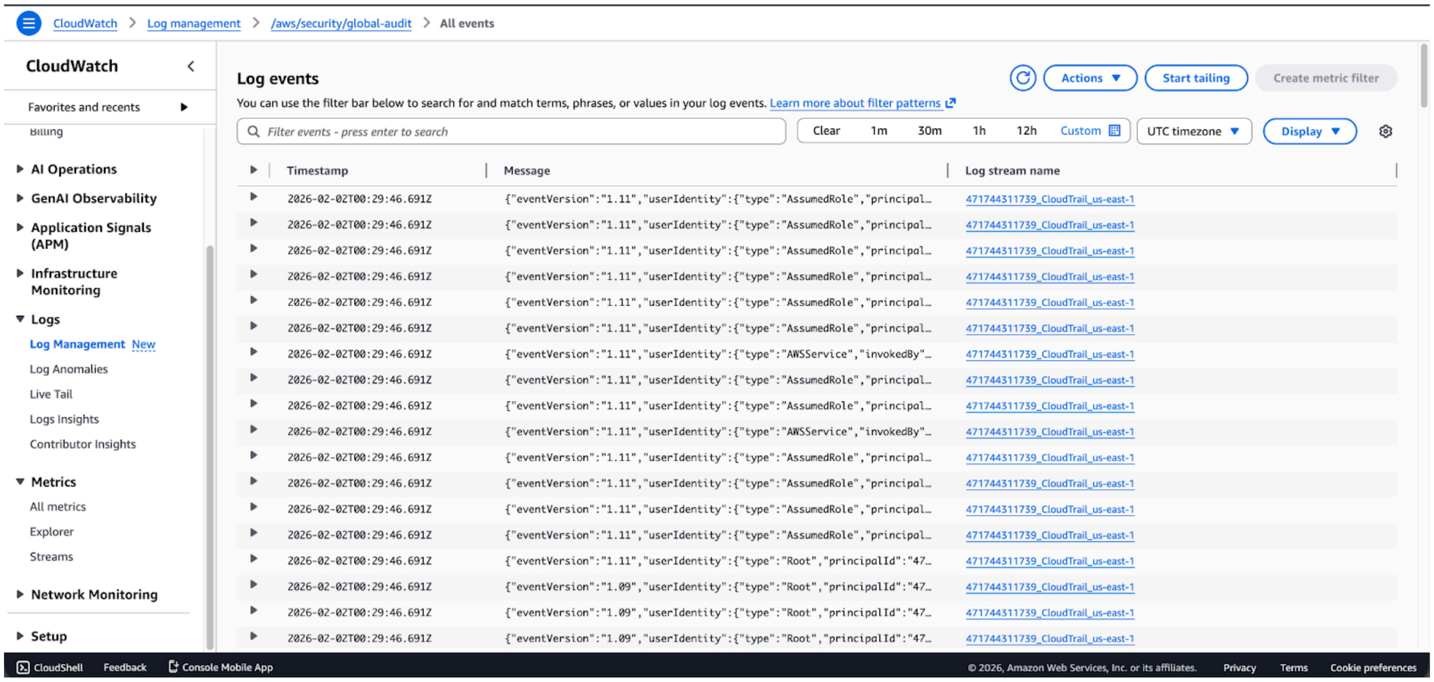
- **Accessed Alarm Dashboard:** Navigated to the CloudWatch Alarms section of the console to see the resource Terraform created.
- **Located the Monitor:** Confirmed that the alarm named `CRITICAL-Root-Login-Detected` is active in the list.
- **Checked Status:** Observed that the alarm is currently in the Insufficient data state, which is normal until the first relevant event (a root login) occurs to trigger a state change.
- **Confirmed Thresholds:** Verified the logic: RootUsage >= 1 for 1 datapoint within 1 minute will trigger the alert.



Name	State	Last state update (UTC)	Conditions	Actions
CRITICAL-Root-Login-Detected	Insufficient data	2026-02-02 00:24:34	RootUsage >= 1 for 1 datapoints within 1 minute	Actions enabled Warning

Step 8: Attack test to trigger the alert

- **Logged Out of IAM:** Fully signed out of the AWS Management Console to clear session as the `terraform-user`.
- **Simulated a High-Risk Event:** Logged back in using Root User credentials, which is the exact behavior Terraform script designed to detect.
- **Generated Traceable Activity:** Navigated to services like S3 or EC2 and performed basic actions to ensure the "Console Login" event was captured by CloudTrail.
- **Observed the Log Delay:** Waited for the necessary 5 to 15 minutes, allowing AWS CloudTrail time to bundle the activity logs and deliver them to CloudWatch.
- **Verified the Email Alert:** Checked inbox for a message from SNS topic, proving that the "Root Login" tripwire successfully triggered the notification system.



You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                   Message                                                                  Log stream name
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AWSService","invokedBy"…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AWSService","invokedBy"…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"AssumedRole","principal…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.11","userIdentity":{"type":"Root","principalId":"47…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.09","userIdentity":{"type":"Root","principalId":"47…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.09","userIdentity":{"type":"Root","principalId":"47…    471744311739_CloudTrail_us-east-1
2026-02-02T00:29:46.691Z    {"eventVersion":"1.09","userIdentity":{"type":"Root","principalId":"47…    471744311739_CloudTrail_us-east-1
```