Big SIEM Energy

At micro-SIEM cost

Kenneth Kaye
JupiterOne

Session ID: SPCNKQ

SIEMs are expensive

- SaaS solutions charge by volume which adds up quickly
- AWS GuardDuty operates similarly
- Build-your-own (looking at you, Matano) introduces other costs!

A cheaper alternative

- AWS CloudTrail +
- AWS EventBridge +
- AWS SNS +
- AWS Chatbot = micro-SIEM!

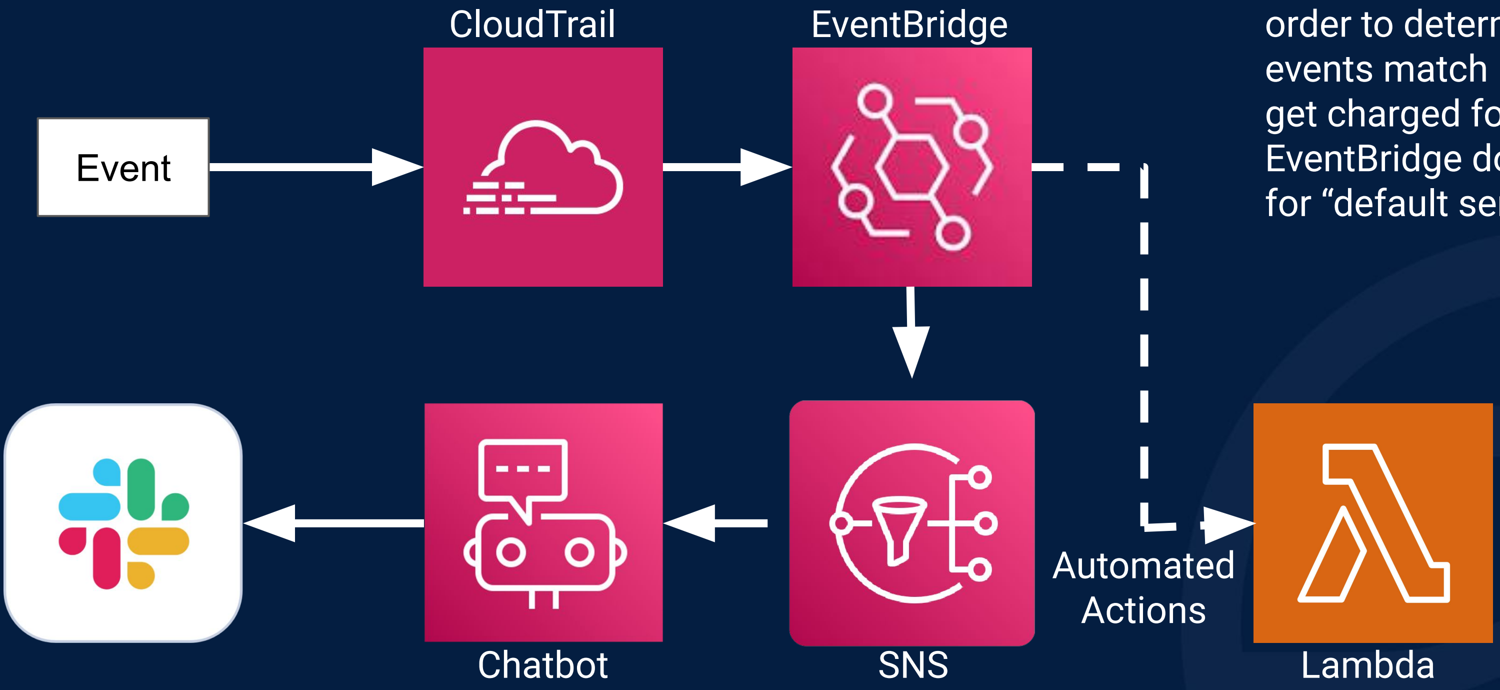
Cloud-native functions and the DIE triad

DIE supplements CIA

- Distributed
- Immutable
- Ephemeral

Lambda functions, containers, and the chaining of service-native functions can fulfill all of those requirements

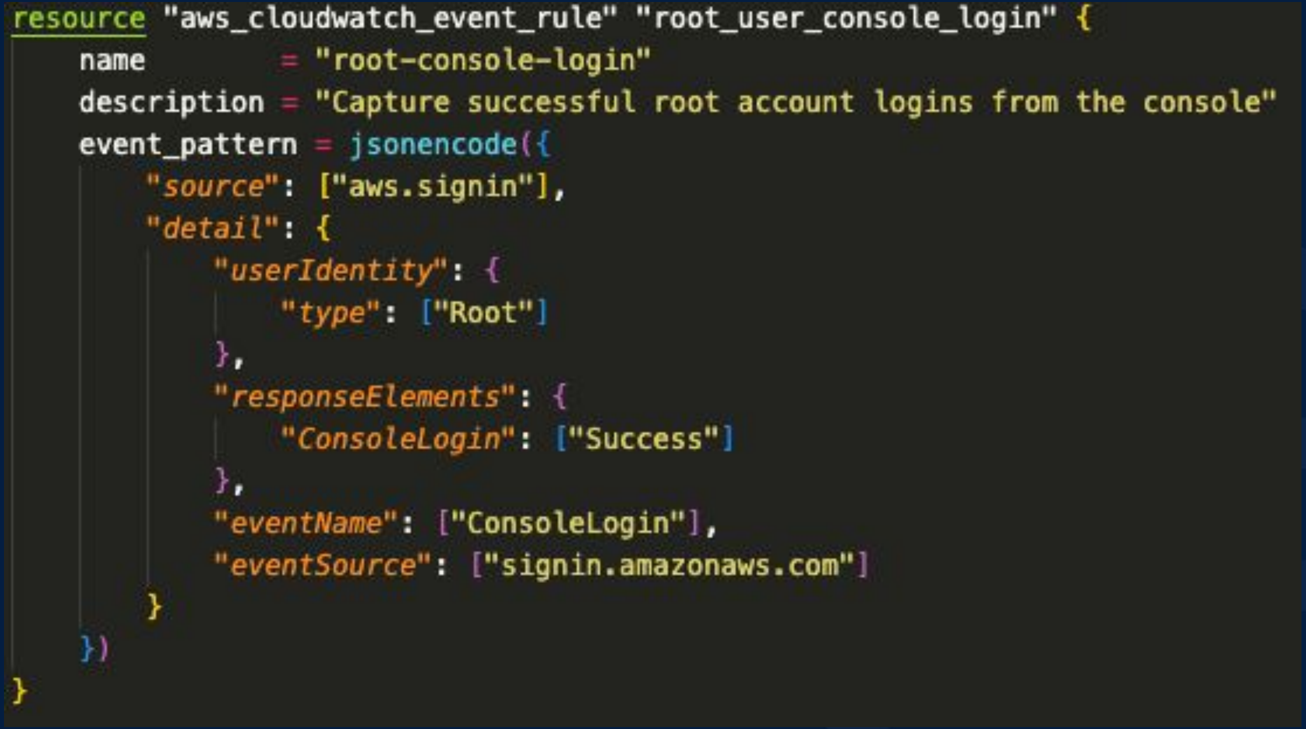
Putting it all together



EventBridge is a little different from how SIEM fires rules, because a SIEM must evaluate all events in order to determine which events match rules and you get charged for every event. EventBridge doesn't charge for "default service events".

Example code in Terraform

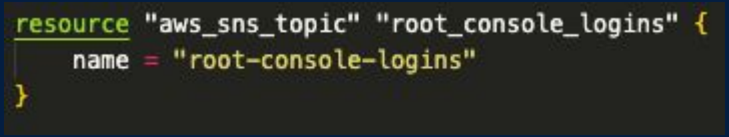
1. Define a rule



```
resource "aws_cloudwatch_event_rule" "root_user_console_login" {
    name          = "root-console-login"
    description = "Capture successful root account logins from the console"
    event_pattern = jsonencode({
        "source": ["aws.signin"],
        "detail": {
            "userIdentity": {
                "type": ["Root"]
            },
            "responseElements": {
                "ConsoleLogin": ["Success"]
            },
            "eventName": ["ConsoleLogin"],
            "eventSource": ["signin.amazonaws.com"]
        }
    })
}
```

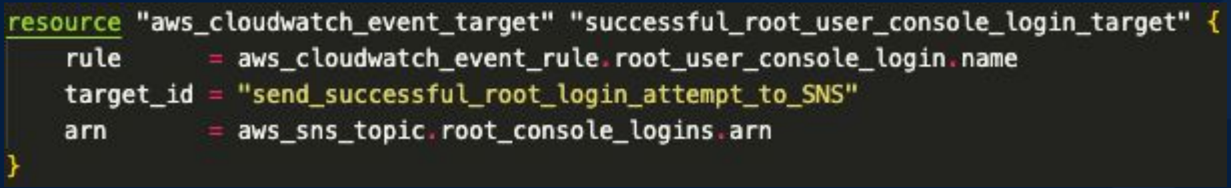
Example code in Terraform (cont)

2. Define a SNS Topic to fire when the rule matches



```
resource "aws_sns_topic" "root_console_logins" {
    name = "root-console-logins"
}
```

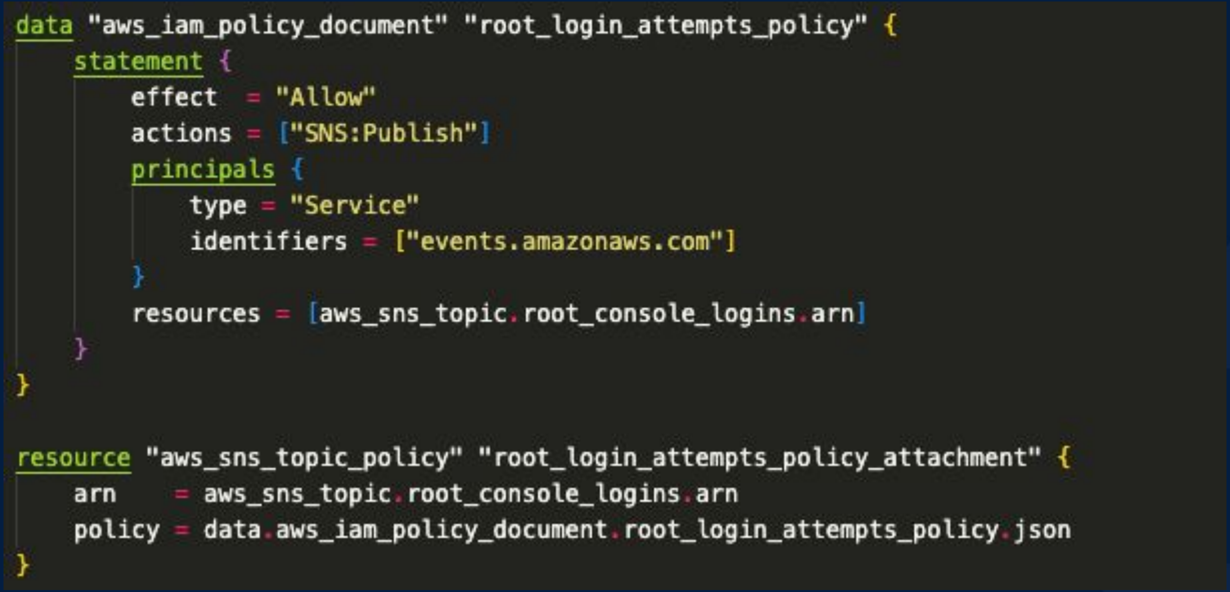
3. Set the SNS Topic as a destination for the EventBridge rule



```
resource "aws_cloudwatch_event_target" "successful_root_user_console_login_target" {
    rule      = aws_cloudwatch_event_rule.root_user_console_login.name
    target_id = "send_successful_root_login_attempt_to_SNS"
    arn       = aws_sns_topic.root_console_logins.arn
}
```

Example code in Terraform (cont)

5. Set minimal permissions to allow the SNS service to write events to that SNS topic

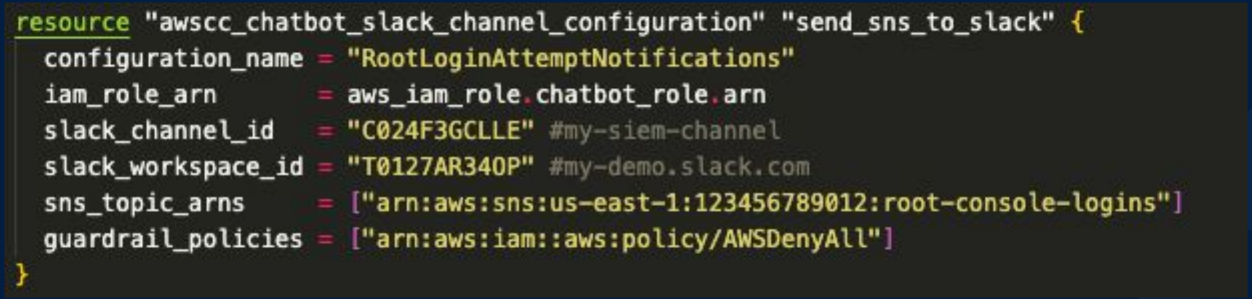


```
data "aws_iam_policy_document" "root_login_attempts_policy" {
    statement {
        effect  = "Allow"
        actions = ["SNS:Publish"]
        principals {
            type = "Service"
            identifiers = ["events.amazonaws.com"]
        }
        resources = [aws_sns_topic.root_console_logins.arn]
    }
}

resource "aws_sns_topic_policy" "root_login_attempts_policy_attachment" {
    arn    = aws_sns_topic.root_console_logins.arn
    policy = data.aws_iam_policy_document.root_login_attempts_policy.json
}
```

Example code in Terraform (cont)

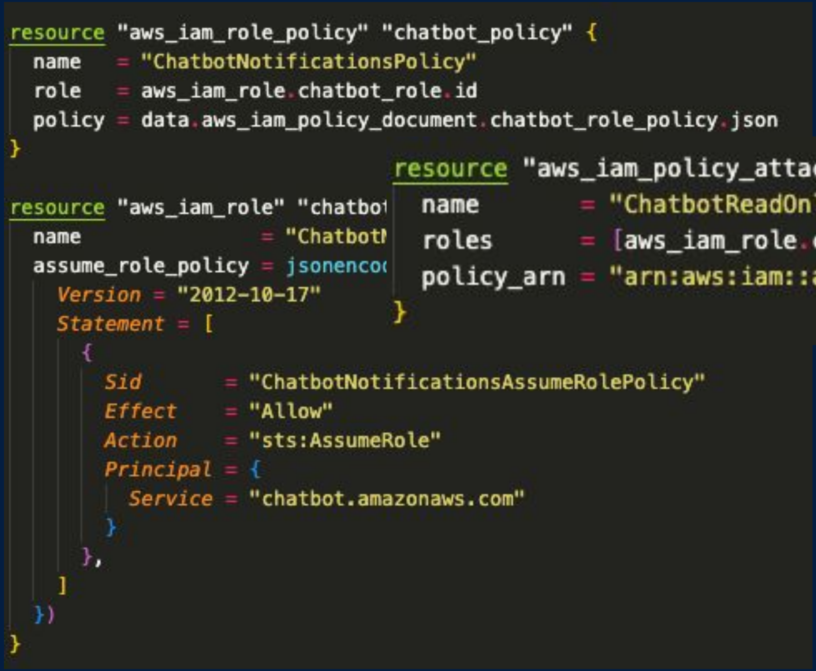
6. The Chatbot service needs to be enabled manually in the AWS Console and linked to your Slack Workspace by a user who can both enable the AWS service and can administer your Slack Workspace.
7. Then create your Chatbot



```
resource "awscc_chatbot_slack_channel_configuration" "send_sns_to_slack" {
  configuration_name = "RootLoginAttemptNotifications"
  iam_role_arn       = aws_iam_role.chatbot_role.arn
  slack_channel_id   = "C024F3GCLLE" #my-siem-channel
  slack_workspace_id = "T0127AR340P" #my-demo.slack.com
  sns_topic_arns     = ["arn:aws:sns:us-east-1:123456789012:root-console-logins"]
  guardrail_policies = ["arn:aws:iam::aws:policy/AWSDenyAll"]
}
```

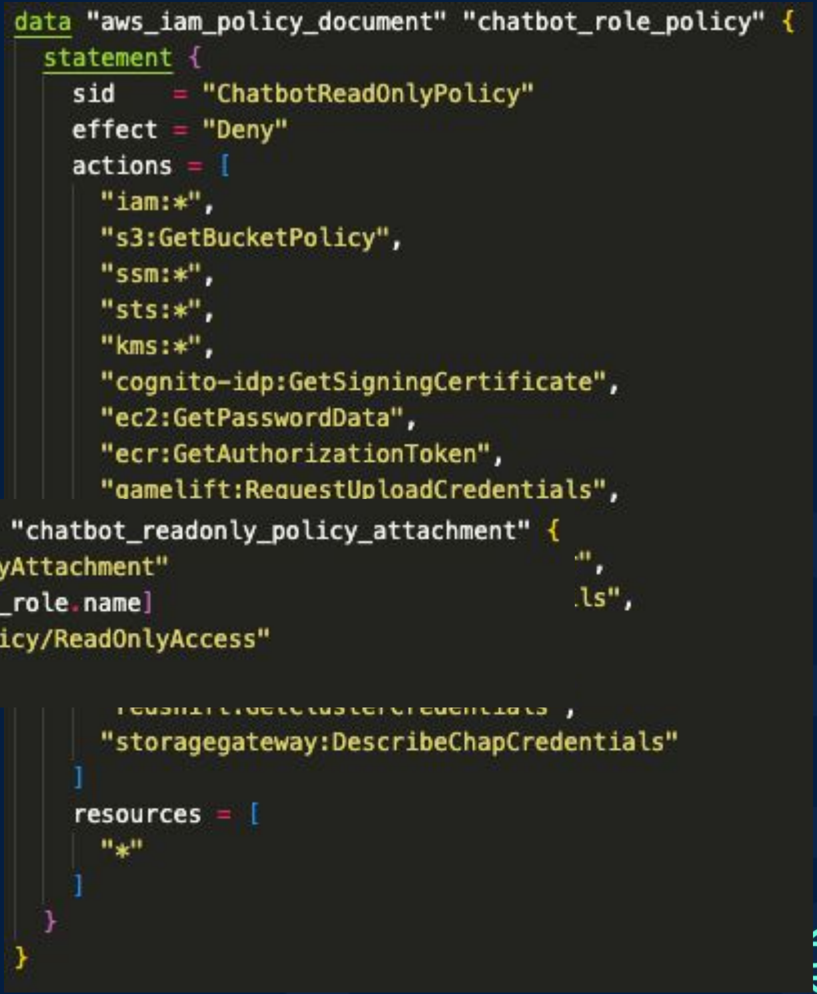
Example code in Terraform (cont)

8. Lock that Chatbot down

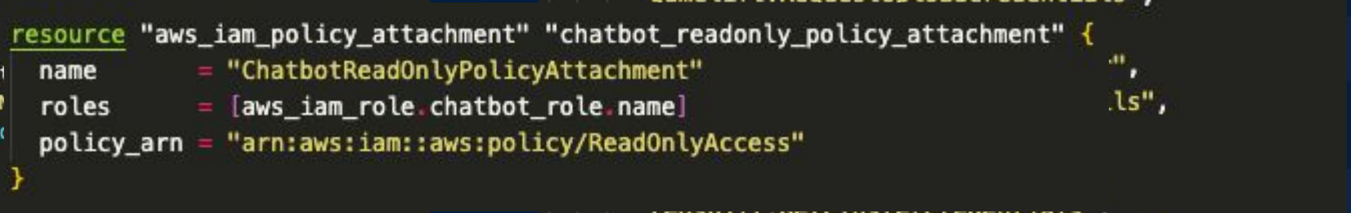


```
resource "aws_iam_role_policy" "chatbot_policy" {
  name   = "ChatbotNotificationsPolicy"
  role   = aws_iam_role.chatbot_role.id
  policy = data.aws_iam_policy_document.chatbot_role_policy.json
}

resource "aws_iam_role" "chatbot
  name                = "ChatbotN
  assume_role_policy = jsonencod
    Version = "2012-10-17"
    Statement = [
      {
        Sid       = "ChatbotNotificationsAssumeRolePolicy"
        Effect    = "Allow"
        Action    = "sts:AssumeRole"
        Principal = {
          Service = "chatbot.amazonaws.com"
        }
      },
    ]
  })
}
```

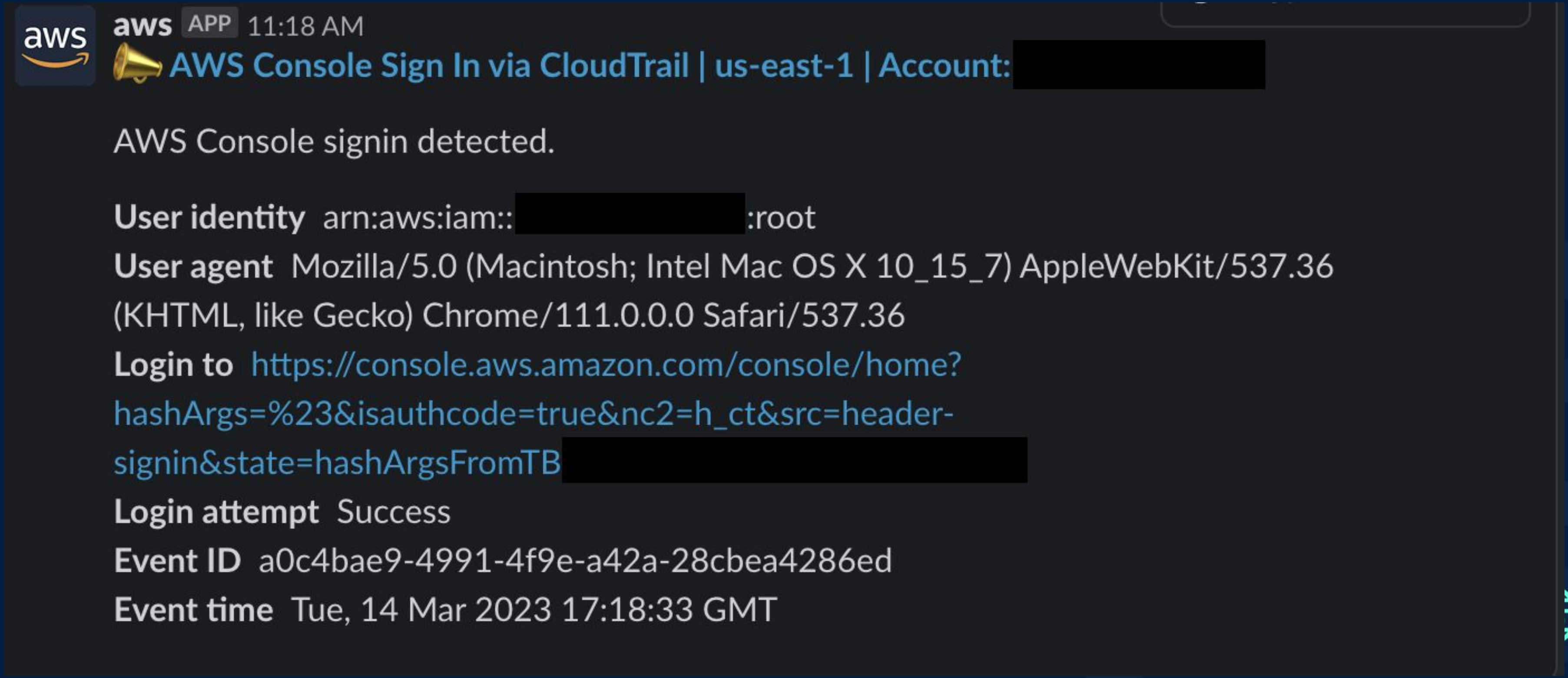


```
data "aws_iam_policy_document" "chatbot_role_policy" {
  statement {
    sid    = "ChatbotReadOnlyPolicy"
    effect = "Deny"
    actions = [
      "iam:*",
      "s3:GetBucketPolicy",
      "ssm:*",
      "sts:*",
      "kms:*",
      "cognito-idp:GetSigningCertificate",
      "ec2:GetPasswordData",
      "ecr:GetAuthorizationToken",
      "gamelift:RequestUploadCredentials",
                                          ",
                                         ls",
      "redshift:GetClusterCredentials",
      "storagegateway:DescribeChapCredentials"
    ]
    resources = [
      "*"
    ]
  }
}
```



```
resource "aws_iam_policy_attachment" "chatbot_readonly_policy_attachment" {
  name       = "ChatbotReadOnlyPolicyAttachment"
  roles      = [aws_iam_role.chatbot_role.name]
  policy_arn = "arn:aws:iam::aws:policy/ReadOnlyAccess"
}
```

Success!



aws APP 11:18 AM

AWS Console Sign In via CloudTrail | us-east-1 | Account: [redacted]

AWS Console signin detected.

User identity arn:aws:iam::[redacted]:root
User agent Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36
Login to https://console.aws.amazon.com/console/home?hashArgs=%23&isauthcode=true&nc2=h_ct&src=header-signin&state=hashArgsFromTB[redacted]
Login attempt Success
Event ID a0c4bae9-4991-4f9e-a42a-28cbea4286ed
Event time Tue, 14 Mar 2023 17:18:33 GMT

Resources

- AWS CloudTrail pricing: https://aws.amazon.com/cloudtrail/pricing/
- AWS EventBridge pricing: https://aws.amazon.com/eventbridge/pricing/
- AWS SNS pricing: https://aws.amazon.com/sns/pricing/
- AWS Chatbot pricing: https://aws.amazon.com/chatbot/pricing/
- AWS GuardDuty pricing: https://aws.amazon.com/guardduty/pricing/
- Terraform documentation for AWS: https://registry.terraform.io/providers/hashicorp/aws/latest/docs
- Average SIEM costs:
  - https://www.acecloudhosting.com/blog/managed-siem-pricing/#:~:text=Generally%2C%20a%20managed%20SIEM%20will,that%20needs%20to%20be%20monitored.
  - https://schoenbaum.medium.com/the-average-siem-deployment-costs-18m-annually-cf576f6c740d
  - Internal research and development at JupiterOne

Questions?

TF code and this presentation can be found in my GH repository here ->