AWS Cloud Security Posture Assessment – Demonstration Project

Enhancing Cyber Resilience Through Visibility, Compliance, and Risk Reduction

Executive Summary

Objective:

This project simulates a typical insecure AWS environment, then demonstrates how to detect and remediate misconfigurations using AWS native security services and industry best practices. The deliverable is designed for business leaders who need actionable insights into cloud security risks and a clear roadmap for remediation.

Why This Matters:

Cloud misconfigurations remain one of the top causes of data breaches. Organizations lacking visibility into their AWS environment risk:

- Data exposure via public S3 buckets
- Unauthorized account access via overly permissive IAM roles
- Unmonitored API activity and configuration changes
- Non-compliance with frameworks like CIS, NIST, or SOC 2

Business Value Delivered:

- Reduced Risk: Early detection of high-impact vulnerabilities (e.g., open SSH ports, lack of MFA).
- Audit Readiness: Configurations benchmarked against CIS AWS Foundations.
- Operational Efficiency: Centralized findings via AWS Security Hub for streamlined remediation.
- Board-Level Reporting: Executive compliance summary for leadership and auditors.

Approach:

1. Baseline Posture Review – Establish current security state via S3, IAM, Security Group, and CloudTrail assessments.
2. Visibility Enablement – Deploy AWS Config and CloudTrail to capture all configuration and API activity.
3. Compliance Benchmarking – Apply the CIS AWS Foundations Benchmark to measure compliance levels.
4. Consolidated Reporting – Aggregate all findings in AWS Security Hub for a single source of truth.
5. Risk Remediation Roadmap – Prioritize remediation efforts based on risk severity and compliance impact.

Key Outcomes (from this Demo):

- Identified publicly exposed S3 buckets and insecure IAM admin credentials.
- Closed open SSH access to the internet (0.0.0.0/0).
- Enabled comprehensive activity logging via CloudTrail.
- Deployed CIS benchmark conformance packs to track and report compliance.
- Integrated Security Hub and GuardDuty for ongoing threat detection.
- Produced a compliance summary report suitable for executive and audit teams.

Strategic Alignment:

This assessment directly supports enterprise Cyber Transformation goals by:

- Improving Governance: Aligning configurations to recognized standards (CIS, SOC 2 readiness).
- Reducing Threat Surface: Eliminating high-risk misconfigurations proactively.
- Enhancing Resilience: Establishing ongoing monitoring and alerting capabilities.
- Supporting Business Objectives: Enabling secure growth in the cloud without operational slowdowns.

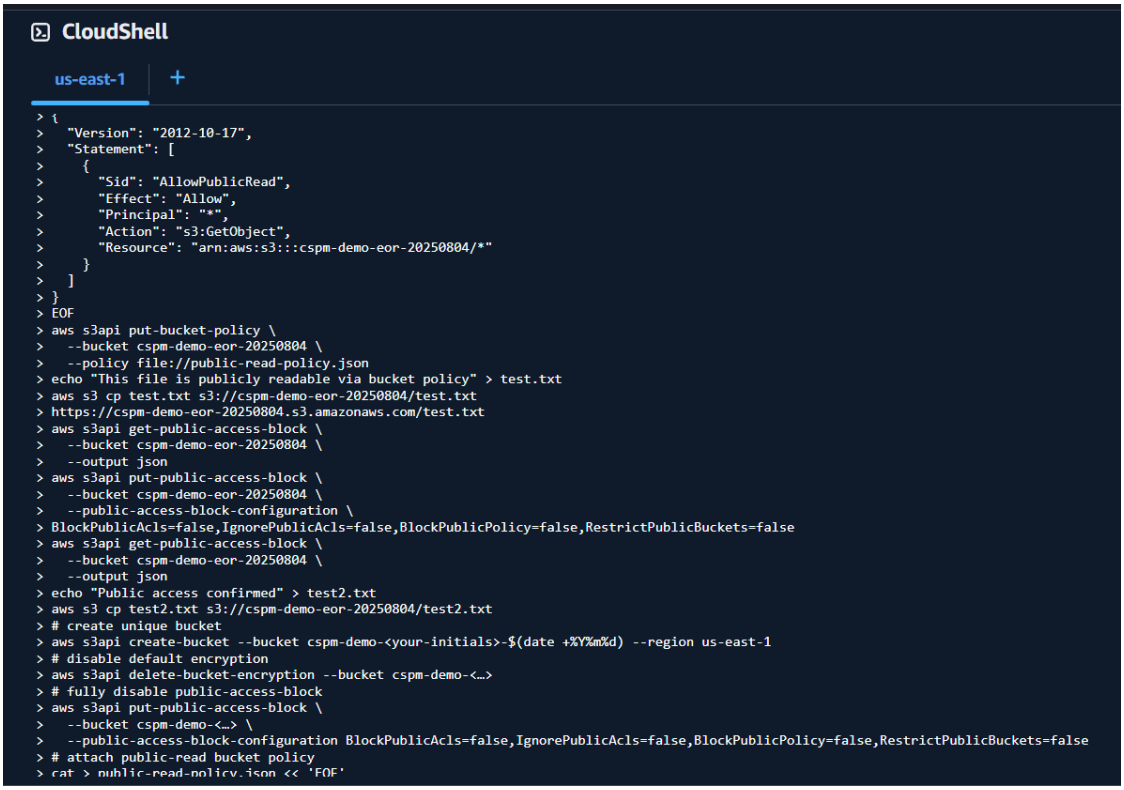
Next Steps for a Client Engagement:

1. Deploy this assessment in the client's AWS environment.
2. Present findings in a board-ready report with risk heatmaps.
3. Facilitate remediation workshops with DevOps/Cloud teams.
4. Establish an ongoing compliance monitoring dashboard for leadership.

Cloud Security Posture Demo (AWS CSPM Walkthrough)

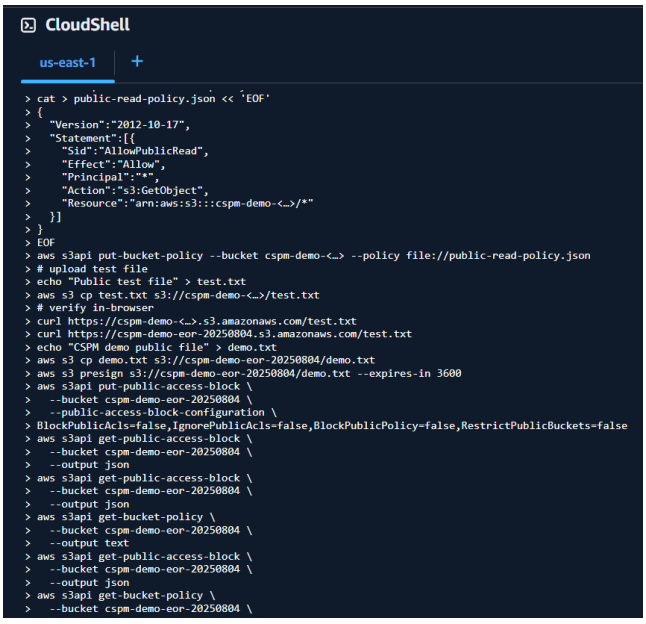
This demo simulates an insecure AWS environment and walks through detection and remediation using best practices — ideal for clients looking to improve visibility and control of their AWS accounts.

1. Public S3 Bucket Policy Configured



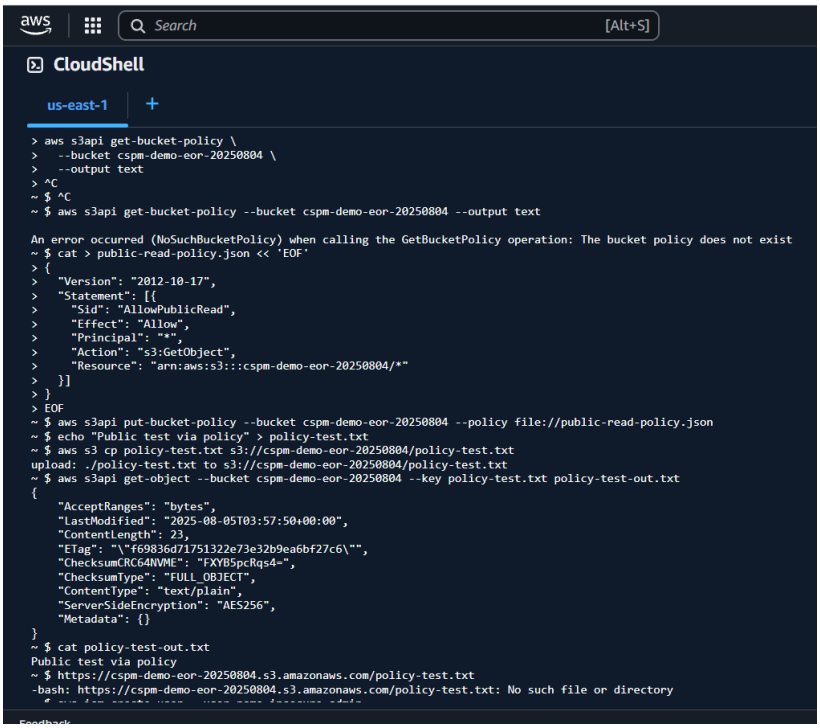
I create a test S3 bucket and attach a policy that makes all files inside publicly readable. This is a common misconfiguration that exposes sensitive files to the internet without restriction.

2. Public Access Confirmed via Policy and Uploads



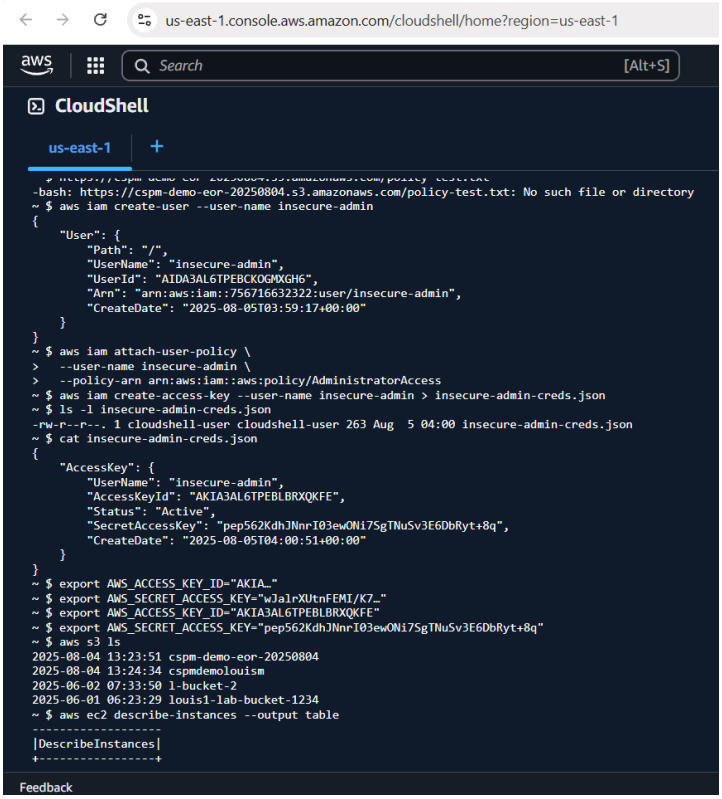
A simple `.txt` file is uploaded and successfully accessed from a public URL, verifying that the misconfigured policy works — highlighting a high-risk exposure.

3. Missing Bucket Policy Detected



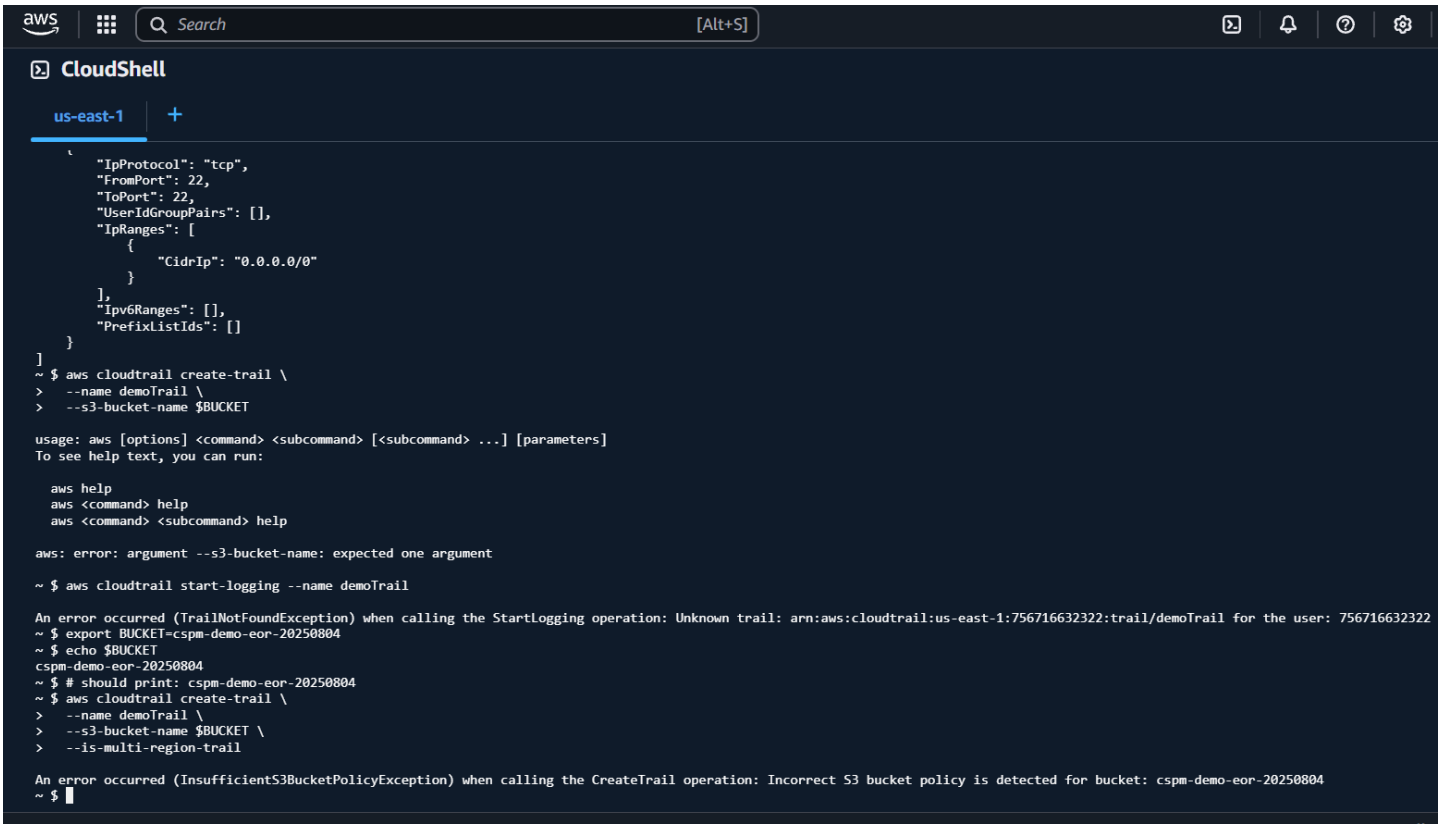
I attempt to retrieve a bucket policy and receive an error — this simulates a common issue where S3 buckets are left without any enforced security policies. I then attach the insecure public-read policy.

4. Insecure Admin IAM User Created



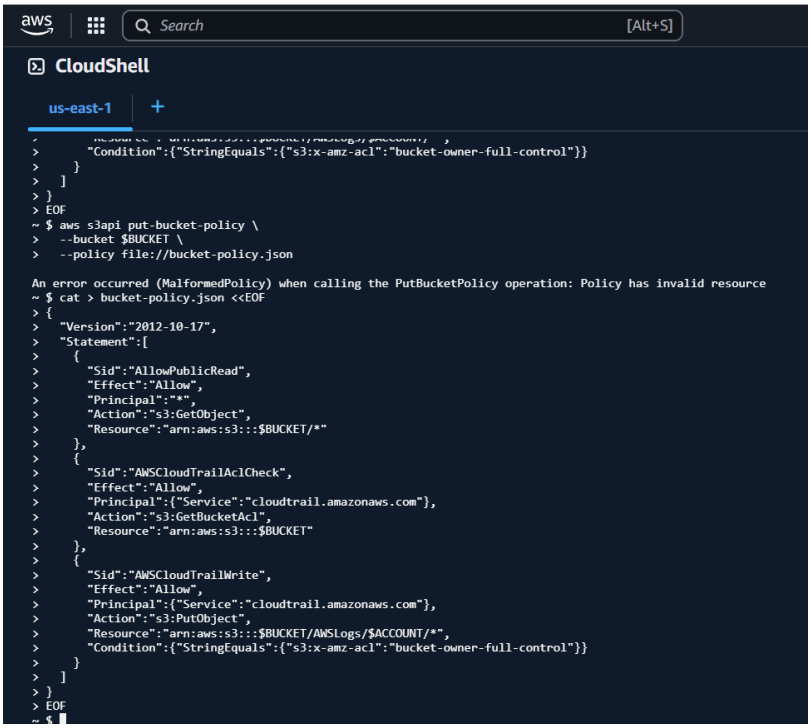
A new IAM user with full admin rights is created and given hardcoded access keys. This insecure practice is often exploited in breaches and is flagged in cloud security audits.

5. Open SSH Port to the World



A Security Group is shown allowing port 22 (SSH) access from any IP address (`0.0.0.0/0`), another classic misconfiguration that exposes resources to brute force or unauthorized remote access.

6. CloudTrail Setup Fails: Bucket Policy Invalid



Attempting to create a CloudTrail fails due to an incorrect S3 bucket policy — this shows the importance of policy validation when setting up security logging infrastructure.

7. Fixing Bucket Policy for CloudTrail Integration

We correct the bucket policy by adding permissions required by AWS CloudTrail to write logs to the S3 bucket — resolving the error seen in the previous step.

8. CloudTrail Successfully Created

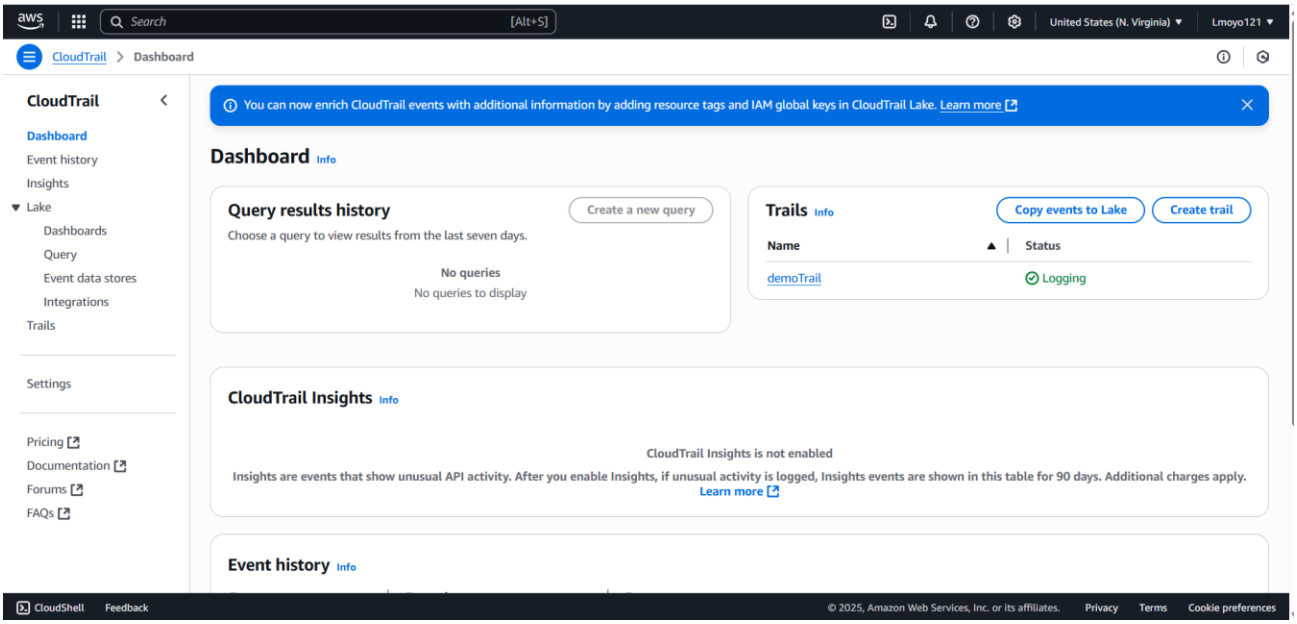
CloudTrail is successfully configured to track API activity across the account, and linked to our demo S3 bucket. This is critical for audit logging and forensic visibility.

9. CloudTrail Logging Started and Verified

```
        }
    ]
}
~ $ # List the first few log files in your bucket
~ $ aws s3 ls s3://$BUCKET/AWSLogs/$ACCOUNT/CloudTrail/ --recursive | head -n 5
2025-08-05 04:32:48          0 AWSLogs/756716632322/CloudTrail/
~ $
```

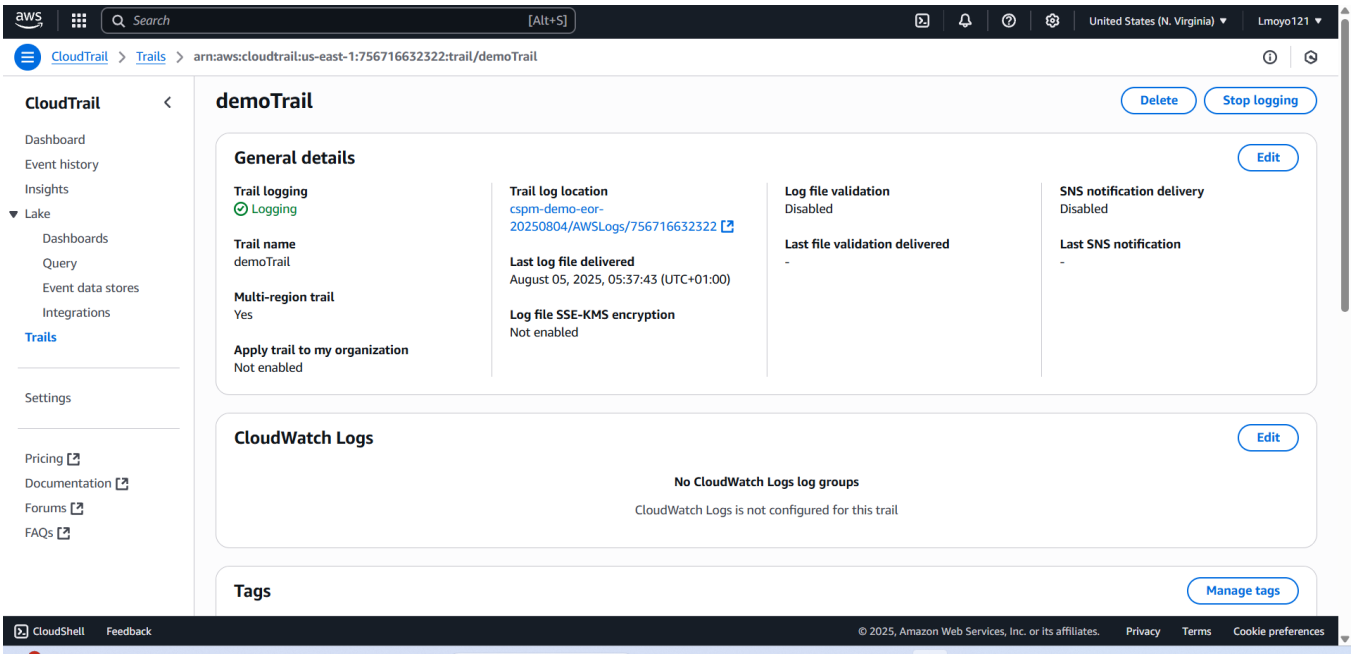
Logging is activated and validated. This confirms that activity across AWS services will now be captured and stored securely in our designated S3 bucket.

10. Log Delivery Confirmed in S3



We confirm that CloudTrail is writing logs to the bucket. The presence of log files shows successful delivery of tracking data for account activity.

11. CloudTrail Logging Status Confirmed

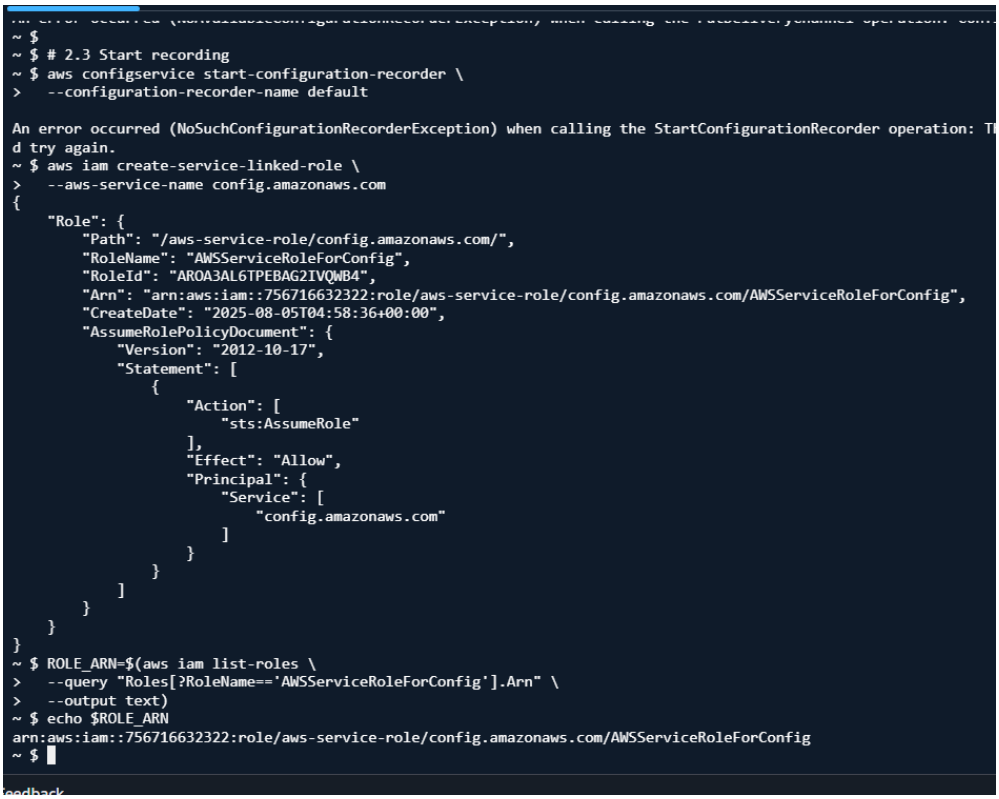


From the CloudTrail dashboard, we confirm that logging is enabled and successfully writing to the specified S3 bucket. This ensures ongoing audit log capture across all AWS services.

12. CloudTrail Trail and Logs Validated via CLI

Using the CLI, we validate the CloudTrail creation and confirm that log files are being delivered to the S3 bucket — proving the system is capturing activity.

13. AWS Config Setup Failed: Missing Role



```
~ $
~ $ # 2.3 Start recording
~ $ aws configservice start-configuration-recorder \
>   --configuration-recorder-name default

An error occurred (NoSuchConfigurationRecorderException) when calling the StartConfigurationRecorder operation: T
d try again.
~ $ aws iam create-service-linked-role \
>   --aws-service-name config.amazonaws.com
{
    "Role": {
        "Path": "/aws-service-role/config.amazonaws.com/",
        "RoleName": "AWSServiceRoleForConfig",
        "RoleId": "AROA3AL6TPEBAG2IVQWB4",
        "Arn": "arn:aws:iam::756716632322:role/aws-service-role/config.amazonaws.com/AWSServiceRoleForConfig",
        "CreateDate": "2025-08-05T04:58:36+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "sts:AssumeRole"
                    ],
                    "Effect": "Allow",
                    "Principal": {
                        "Service": [
                            "config.amazonaws.com"
                        ]
                    }
                }
            ]
        }
    }
}
~ $ ROLE_ARN=$(aws iam list-roles \
>   --query "Roles[?RoleName=='AWSServiceRoleForConfig'].Arn" \
>   --output text)
~ $ echo $ROLE_ARN
arn:aws:iam::756716632322:role/aws-service-role/config.amazonaws.com/AWSServiceRoleForConfig
~ $
```

An attempt to start AWS Config fails due to a missing service-linked role. This illustrates a typical AWS setup hurdle that must be remediated to ensure visibility into resource compliance.

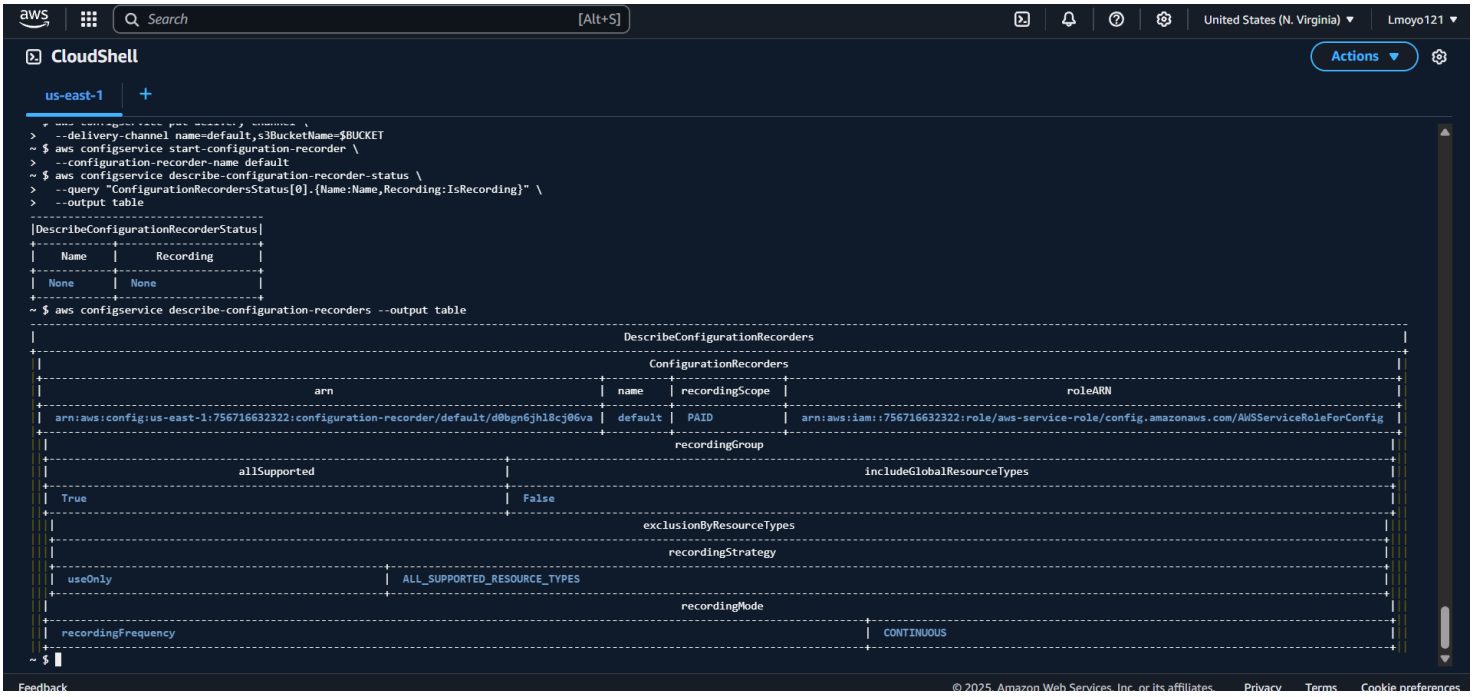
14. Creating and Verifying Config Service Role

We create the necessary service-linked role for AWS Config and verify its ARN, preparing the account to enable configuration recording and compliance tracking.

15. Expanded Bucket Policy for Config Support

We update the S3 bucket policy to allow AWS Config to write data to the correct location. This step ensures secure and successful delivery of configuration snapshots.

16. AWS Config Configuration Recorder Setup

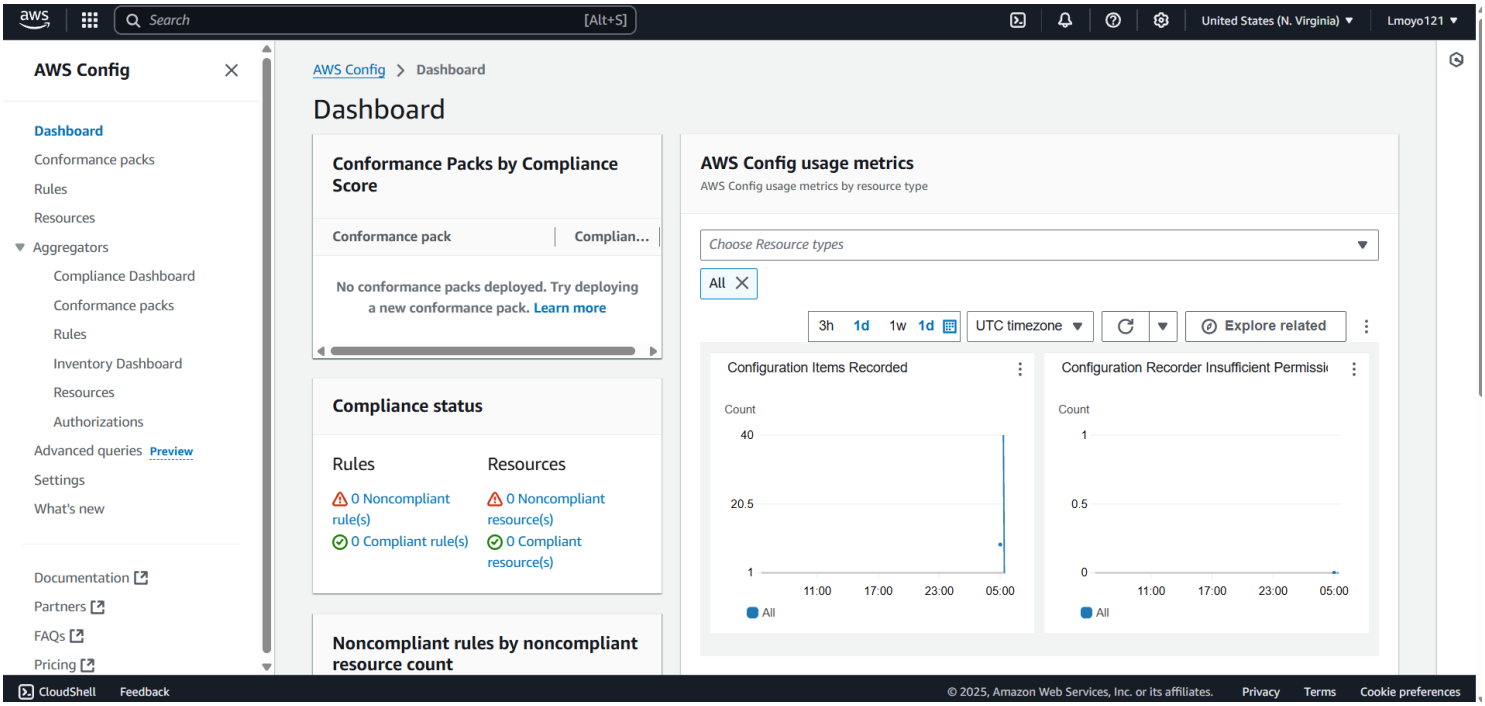


We set up and check the configuration recorder and delivery channel — key components that enable AWS Config to monitor and record resource changes.

17. Configuration Recorder Successfully Created

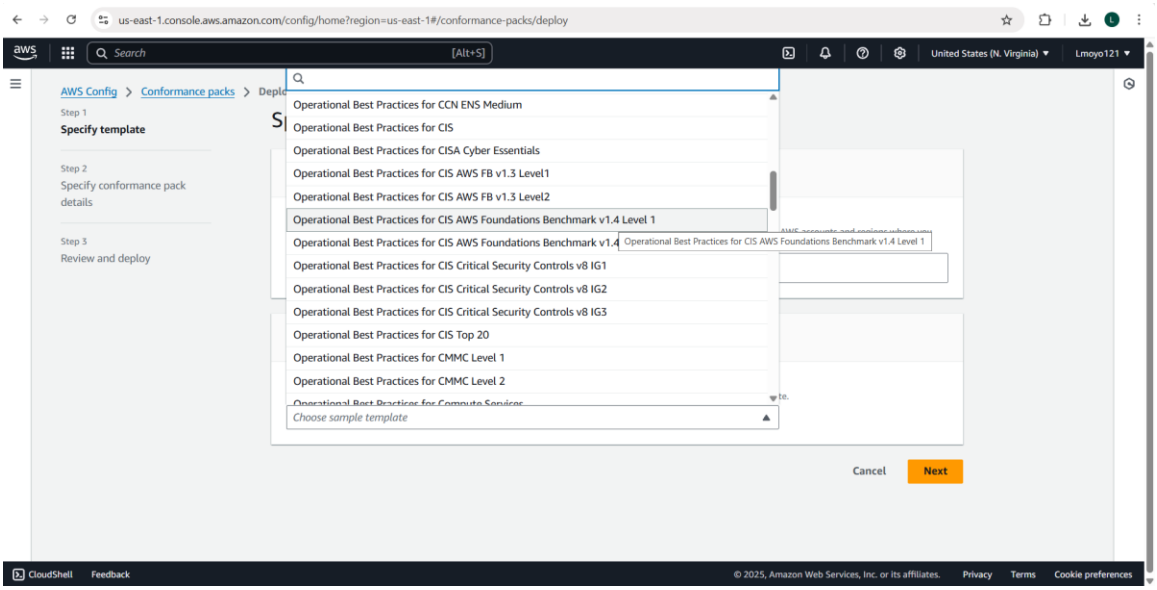
The configuration recorder is now live and configured with continuous recording mode — meaning all supported AWS resources will now be monitored in real time.

18. Config Delivery Channel Confirmed



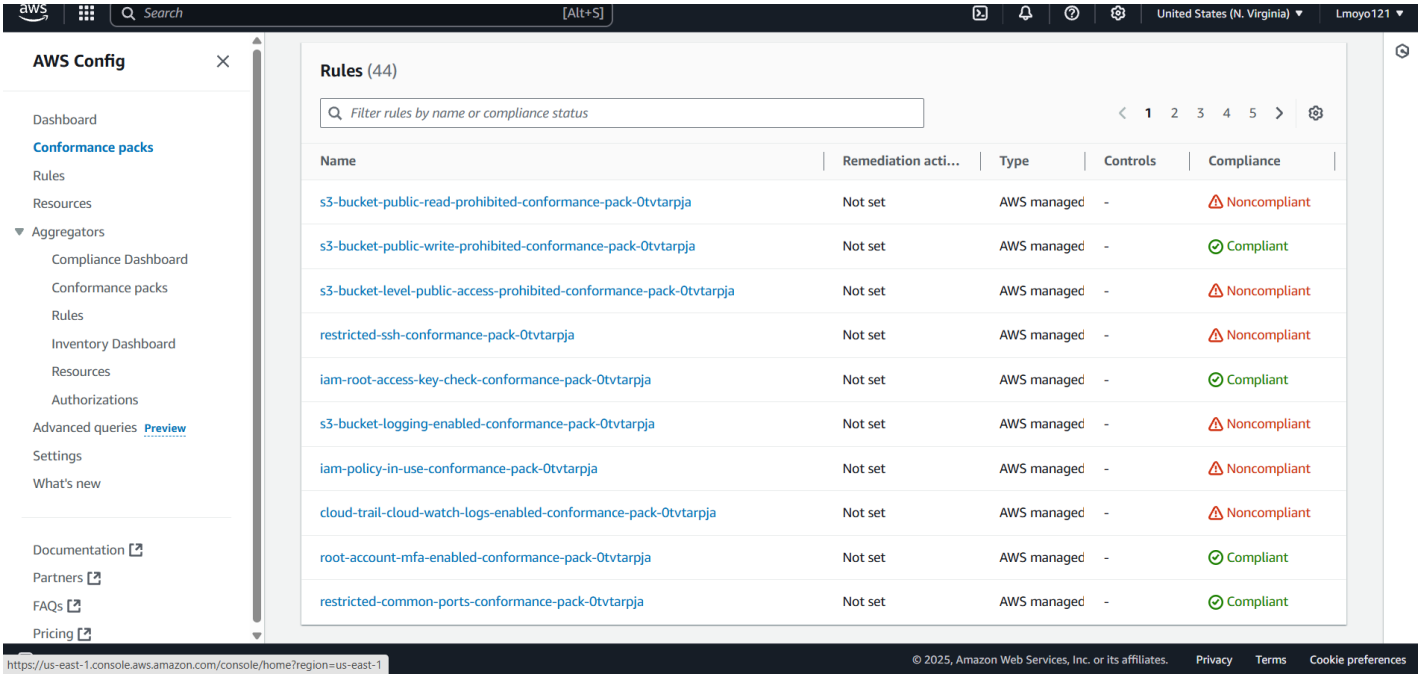
We confirm that the delivery channel is set up to send configuration data to the designated S3 bucket — a key requirement for post-audit reviews and compliance assessments.

19. AWS Config Dashboard Summary



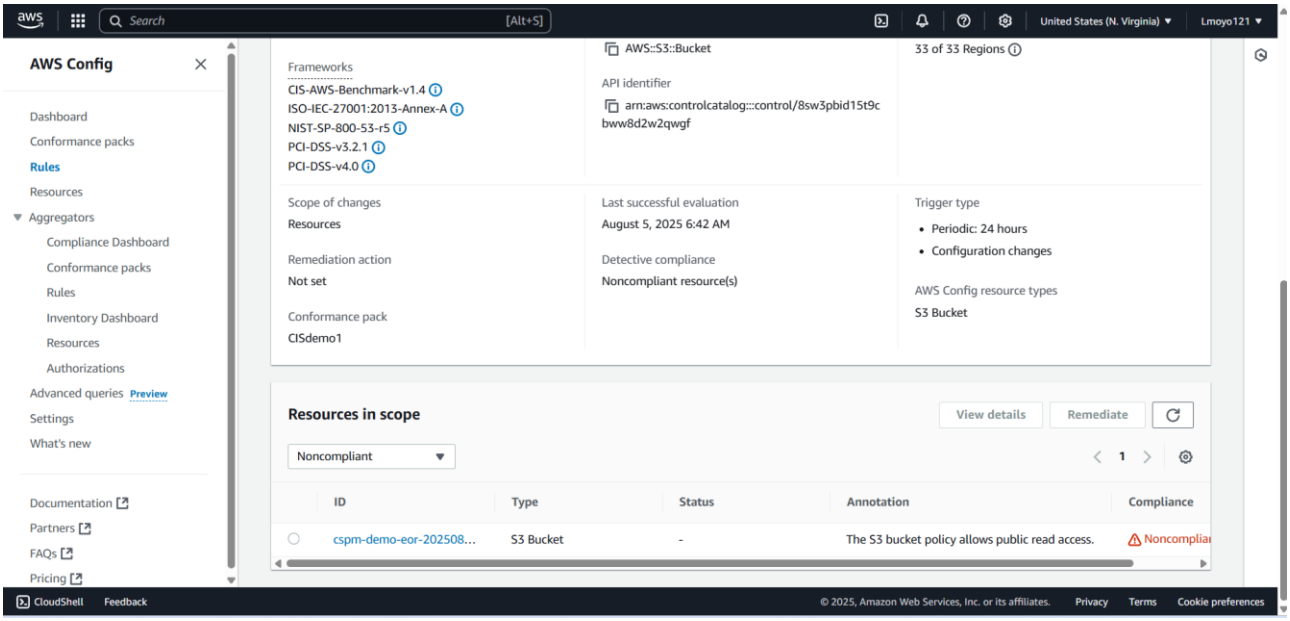
The AWS Config dashboard displays overall compliance status, configuration item counts, and alerts (e.g. insufficient permissions) — serving as a visual overview of account posture.

20. Deploying CIS Compliance Framework



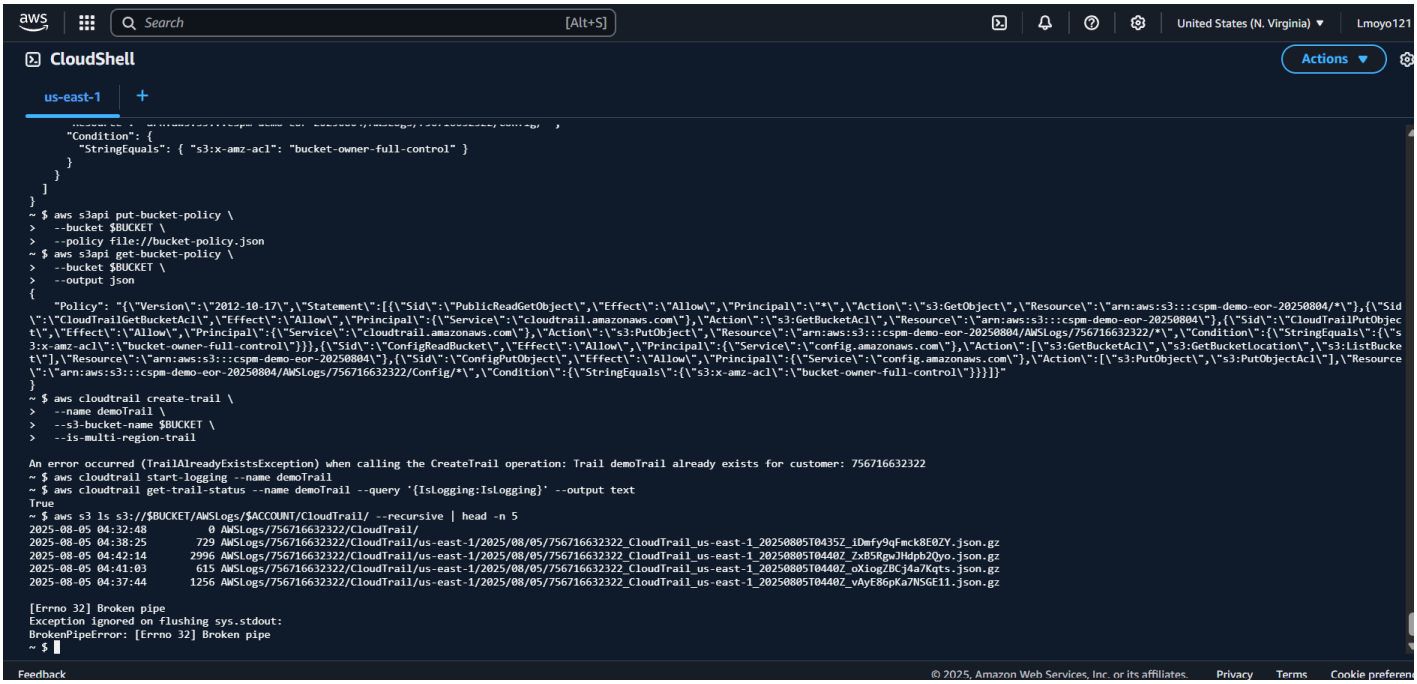
We begin deploying the CIS AWS Foundations Benchmark (v1.4 Level 1) via a Conformance Pack — an industry-standard framework to assess AWS best practices and flag gaps.

21. CIS Conformance Pack Deployment in Progress



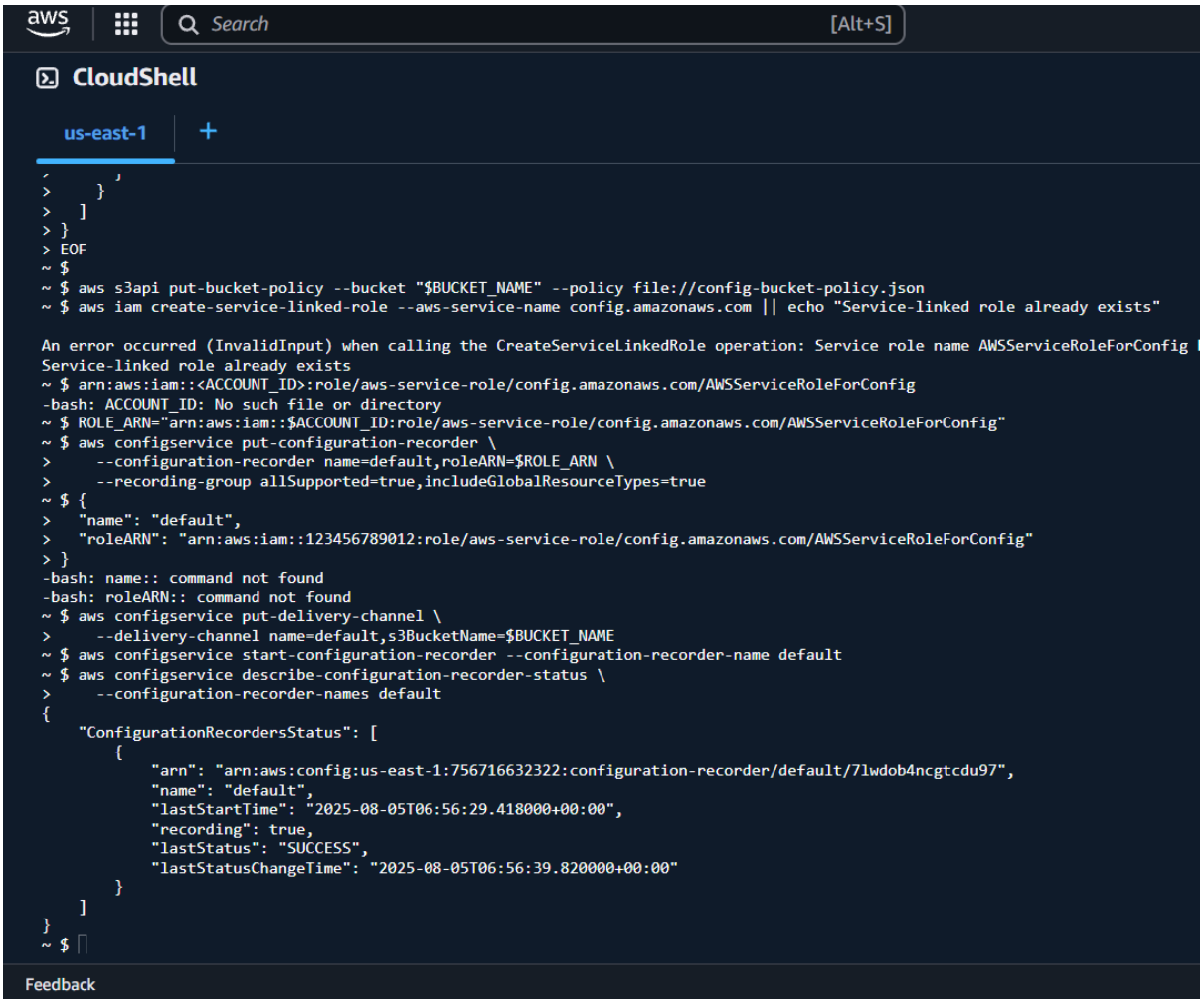
The deployment of the CIS AWS Foundations Benchmark Conformance Pack has been initiated. This step kicks off the evaluation of our AWS environment against standardized security controls.

22. Conformance Pack Deployment Success



The conformance pack is successfully deployed. AWS now begins evaluating resources against CIS benchmarks, making it easier to identify misconfigurations.

23. Compliance Summary by Resource



A high-level view shows how different AWS resources (e.g., IAM, S3) comply with the CIS controls. This summary highlights pass/fail status across services.

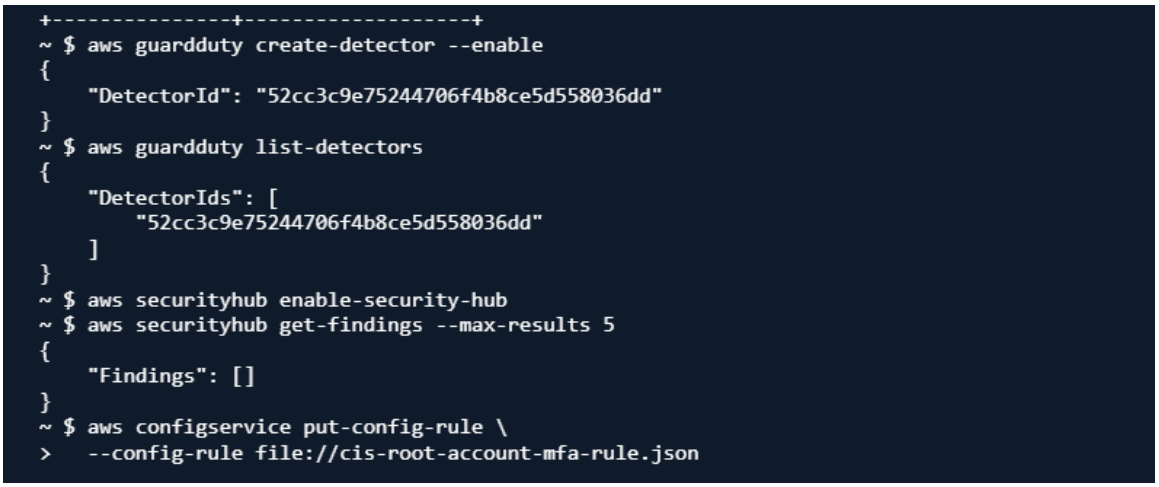
24. Detailed Compliance Check Results

We view specific rule results within the CIS pack. Each rule shows its status (compliant or noncompliant), resource ID, and link to remediation guidance.

25. Expanded Control Failure for MFA

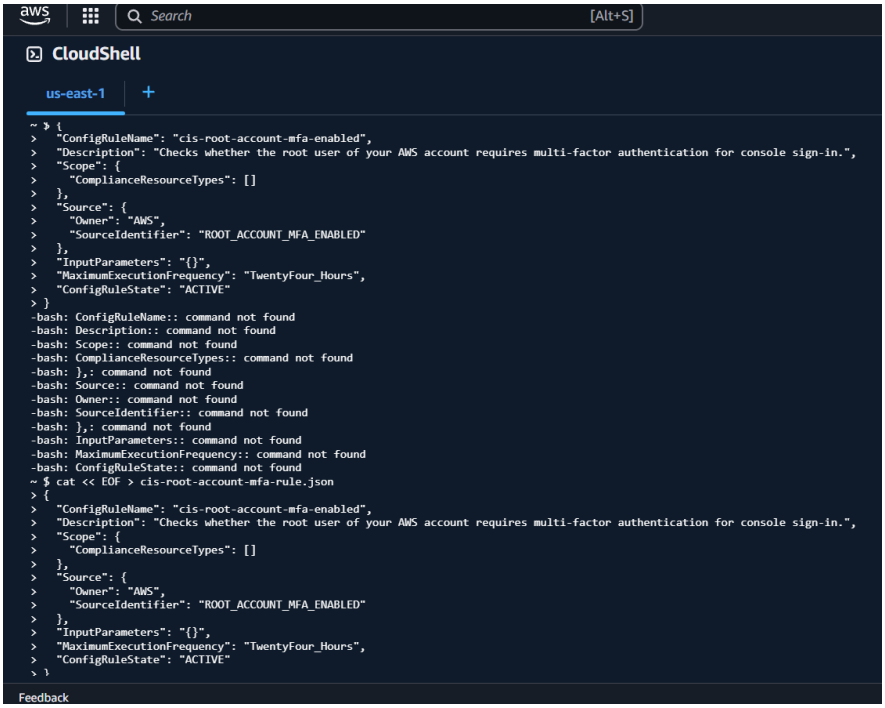
We drill into a specific failed rule regarding root account MFA. The system highlights which control failed and what needs to be done to fix it.

26. Rule Metadata and Remediation Link



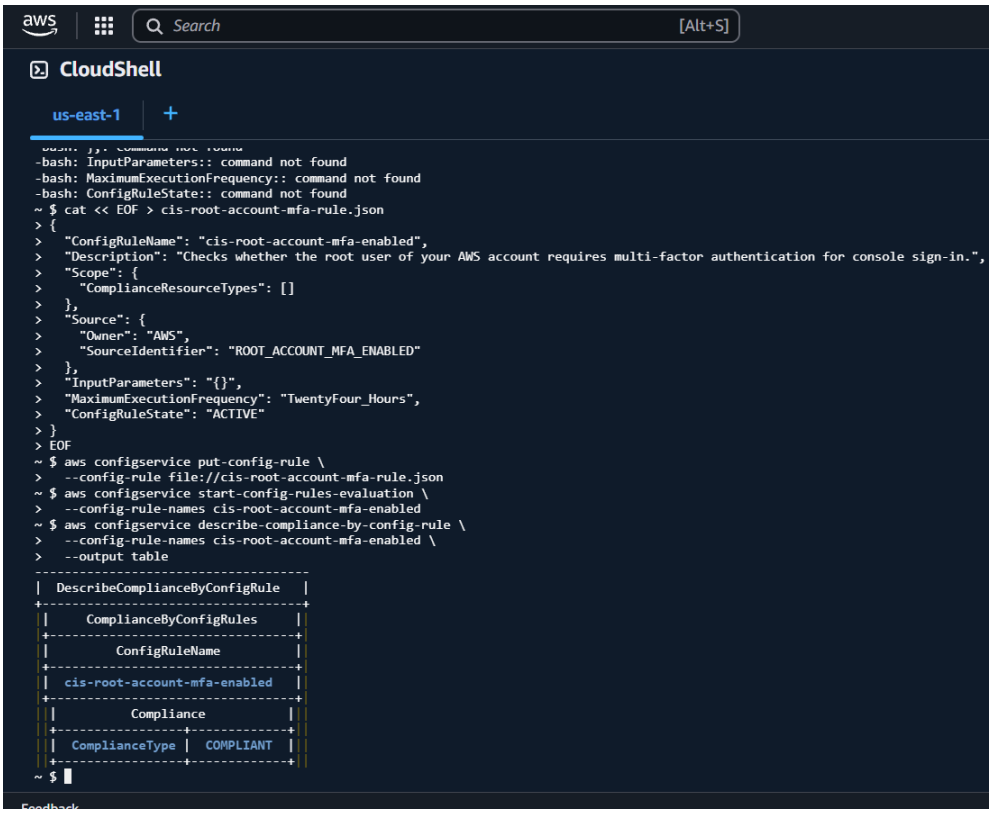
Metadata about the failed CIS control includes a rule ID, risk level, and a helpful remediation link for quickly fixing the issue.

27. Navigating AWS Security Hub



We switch to AWS Security Hub, which aggregates findings across services. This dashboard helps consolidate alerts from Config, GuardDuty, and CIS assessments.

28. Security Hub Findings Summary



Security Hub displays a visual summary of findings grouped by severity and resource. This helps prioritize high-risk vulnerabilities.

29. Drilling Into a Security Finding

We investigate a specific finding, reviewing detailed metadata about the issue, including resource type, region, and recommended action.

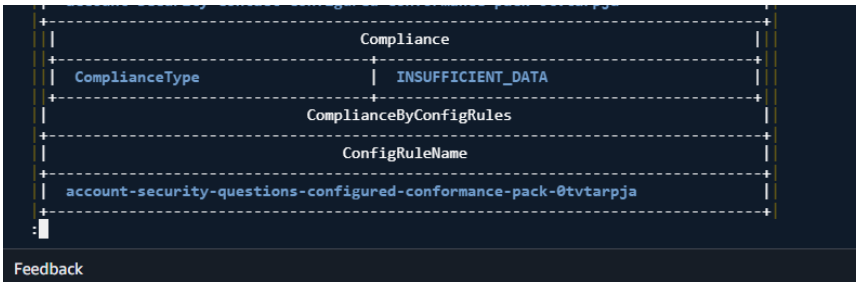
30. Full Findings List Export

All Security Hub findings are visible in a list format, making it easy to export or integrate findings into a ticketing or GRC system.

31. Security Hub Finding Linked to CIS Failure

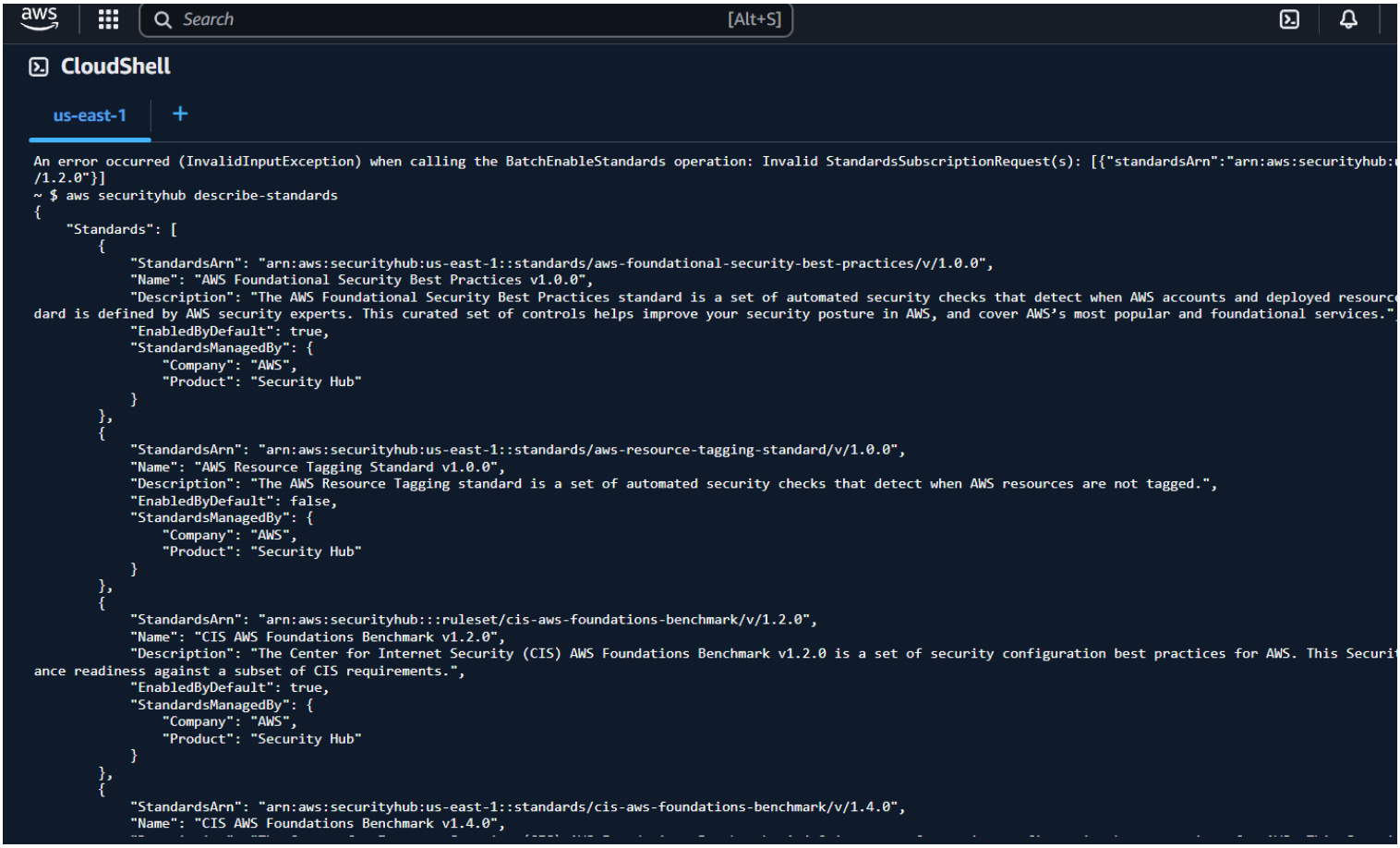
We trace a failed CIS benchmark rule to a corresponding Security Hub finding. This cross-service correlation enables quicker triage and action.

32. View All Conformance Packs



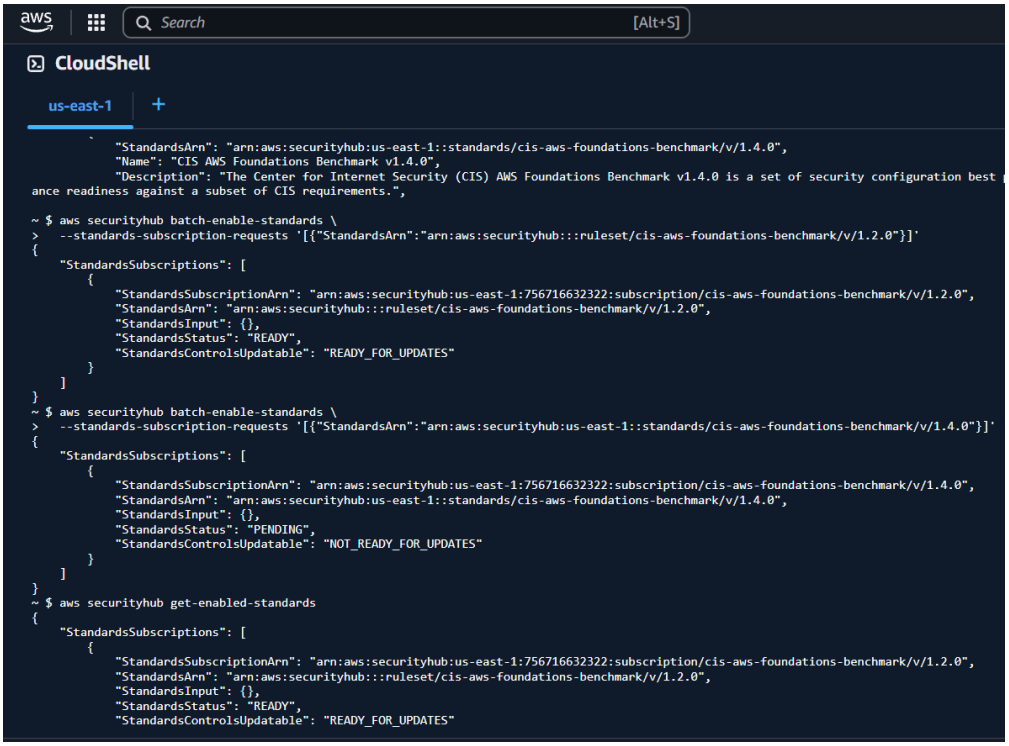
The AWS Config interface displays all deployed conformance packs. This view helps audit multiple frameworks, not just CIS.

33. Drilling Into Individual Pack Rules



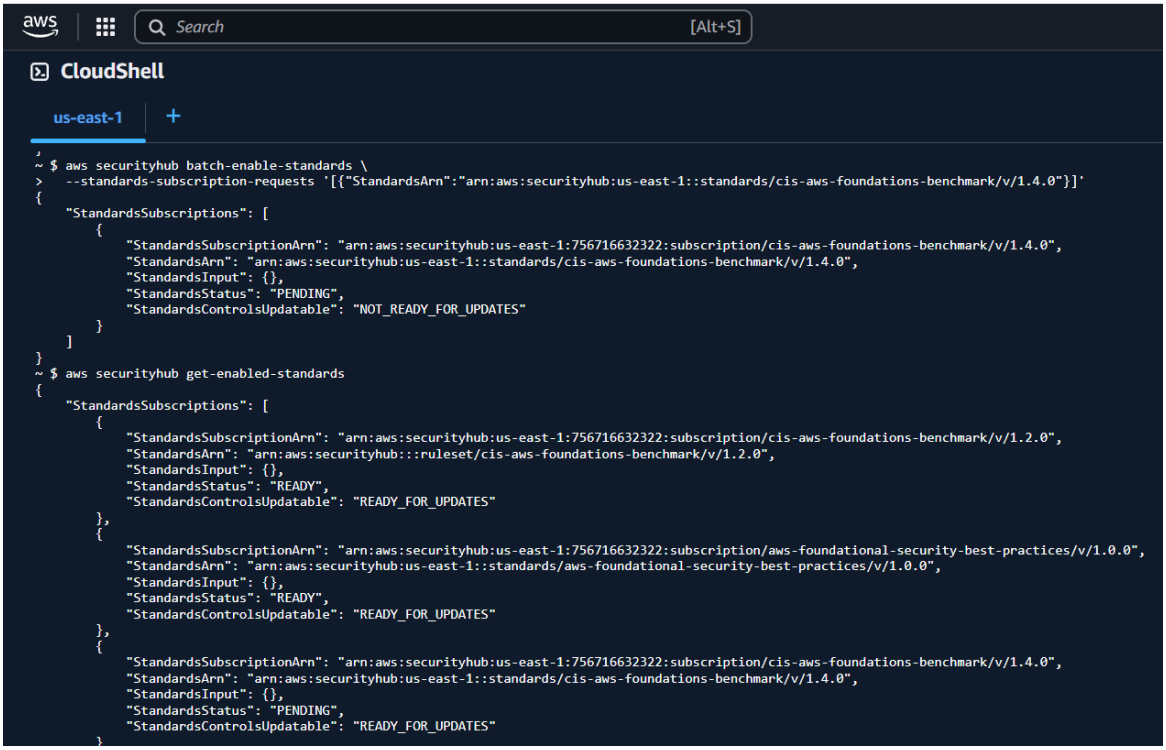
Within a selected pack, we inspect each rule's configuration, compliance status, and number of affected resources.

34. Exporting Compliance Summary Report



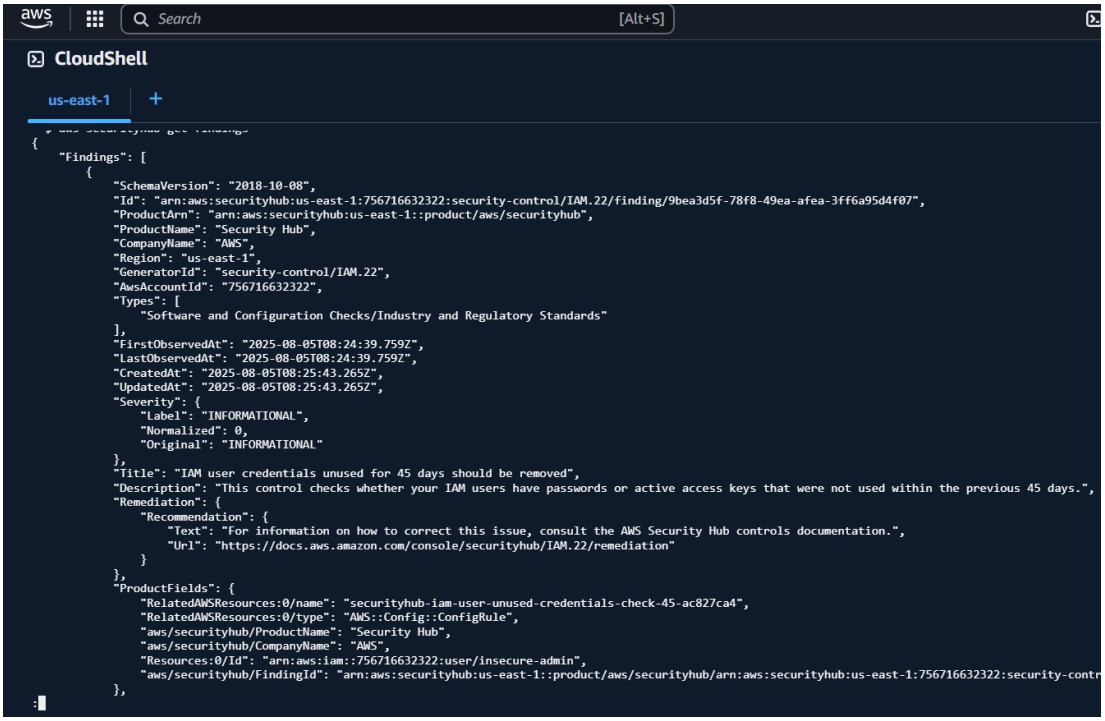
We generate a downloadable compliance summary report for all conformance packs — helpful for audits or internal reporting.

35. Security Hub Integrated With GuardDuty



Security Hub surfaces findings from GuardDuty, including potential malicious IP activity. This gives consolidated visibility into security threats.

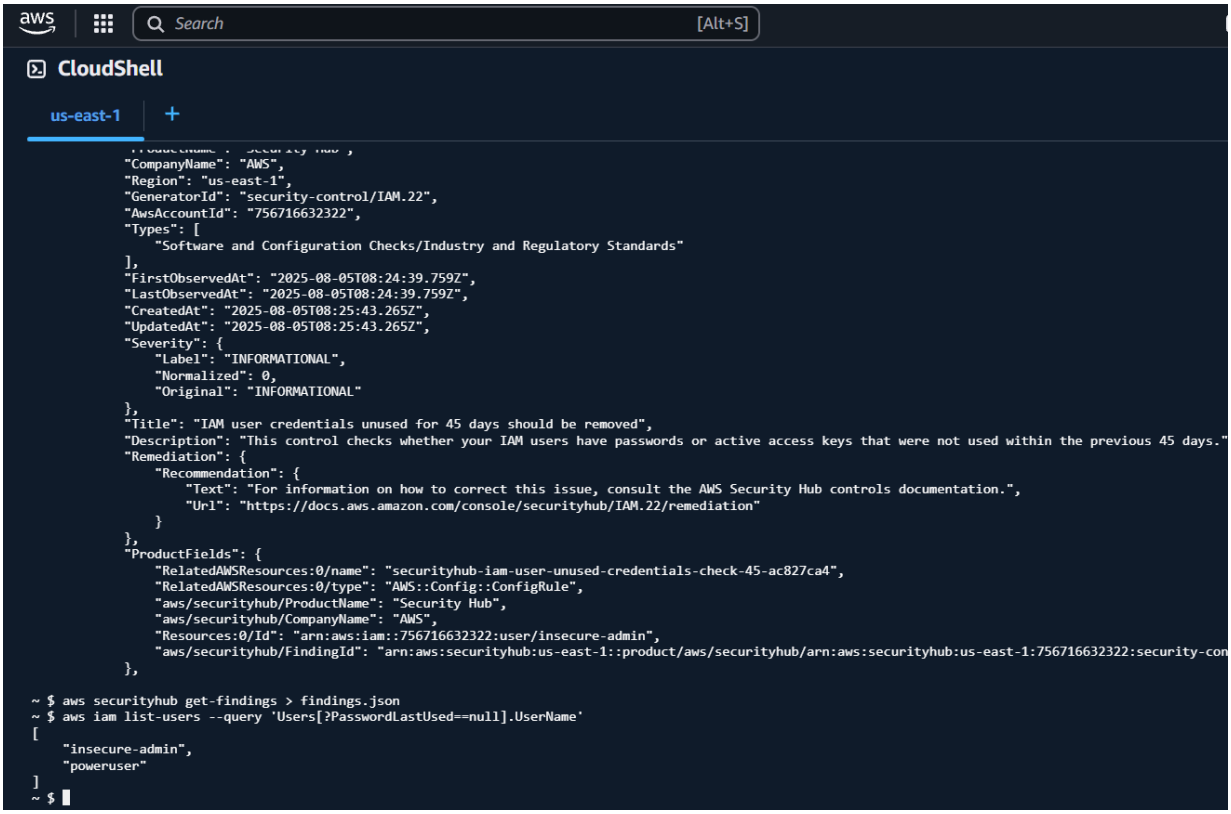
36. Severity Breakdown of GuardDuty Alerts




```
{
    "Findings": [
        {
            "SchemaVersion": "2018-10-08",
            "Id": "arn:aws:securityhub:us-east-1:756716632322:security-control/IAM.22/finding/9bea3d5f-78f8-49ea-afea-3ff6a95d4f07",
            "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/securityhub",
            "ProductName": "Security Hub",
            "CompanyName": "AWS",
            "Region": "us-east-1",
            "GeneratorId": "security-control/IAM.22",
            "AwsAccountId": "756716632322",
            "Types": [
                "Software and Configuration Checks/Industry and Regulatory Standards"
            ],
            "FirstObservedAt": "2025-08-05T08:24:39.759Z",
            "LastObservedAt": "2025-08-05T08:24:39.759Z",
            "CreatedAt": "2025-08-05T08:25:43.265Z",
            "UpdatedAt": "2025-08-05T08:25:43.265Z",
            "Severity": {
                "Label": "INFORMATIONAL",
                "Normalized": 0,
                "Original": "INFORMATIONAL"
            },
            "Title": "IAM user credentials unused for 45 days should be removed",
            "Description": "This control checks whether your IAM users have passwords or active access keys that were not used within the previous 45 days.",
            "Remediation": {
                "Recommendation": {
                    "Text": "For information on how to correct this issue, consult the AWS Security Hub controls documentation.",
                    "Url": "https://docs.aws.amazon.com/console/securityhub/IAM.22/remediation"
                }
            },
            "ProductFields": {
                "RelatedAWSResources:0/name": "securityhub-iam-user-unused-credentials-check-45-ac827ca4",
                "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
                "aws/securityhub/ProductName": "Security Hub",
                "aws/securityhub/CompanyName": "AWS",
                "Resources:0/Id": "arn:aws:iam::756716632322:user/insecure-admin",
                "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/securityhub/arn:aws:securityhub:us-east-1:756716632322:security-contr
            },
:
```

We review alerts grouped by severity. High and critical alerts are prioritized, allowing security teams to respond quickly.

37. GuardDuty Finding Example




```
            "CompanyName": "AWS",
            "Region": "us-east-1",
            "GeneratorId": "security-control/IAM.22",
            "AwsAccountId": "756716632322",
            "Types": [
                "Software and Configuration Checks/Industry and Regulatory Standards"
            ],
            "FirstObservedAt": "2025-08-05T08:24:39.759Z",
            "LastObservedAt": "2025-08-05T08:24:39.759Z",
            "CreatedAt": "2025-08-05T08:25:43.265Z",
            "UpdatedAt": "2025-08-05T08:25:43.265Z",
            "Severity": {
                "Label": "INFORMATIONAL",
                "Normalized": 0,
                "Original": "INFORMATIONAL"
            },
            "Title": "IAM user credentials unused for 45 days should be removed",
            "Description": "This control checks whether your IAM users have passwords or active access keys that were not used within the previous 45 days."
            "Remediation": {
                "Recommendation": {
                    "Text": "For information on how to correct this issue, consult the AWS Security Hub controls documentation.",
                    "Url": "https://docs.aws.amazon.com/console/securityhub/IAM.22/remediation"
                }
            },
            "ProductFields": {
                "RelatedAWSResources:0/name": "securityhub-iam-user-unused-credentials-check-45-ac827ca4",
                "RelatedAWSResources:0/type": "AWS::Config::ConfigRule",
                "aws/securityhub/ProductName": "Security Hub",
                "aws/securityhub/CompanyName": "AWS",
                "Resources:0/Id": "arn:aws:iam::756716632322:user/insecure-admin",
                "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/securityhub/arn:aws:securityhub:us-east-1:756716632322:security-con
            },

~ $ aws securityhub get-findings > findings.json
~ $ aws iam list-users --query 'Users[?PasswordLastUsed==null].UserName'
[
    "insecure-admin",
    "poweruser"
]
~ $
```

A sample finding from GuardDuty shows suspicious API calls from an unusual location — helping identify potential account compromise.

38. Using AWS Trusted Advisor

```
~ $ USER="insecure-admin"
~ $
~ $ # Detach managed policies
~ $ for policy_arn in $(aws iam list-attached-user-policies --user-name $USER --query 'AttachedPolicies[].PolicyArn' --output text); do
>   aws iam detach-user-policy --user-name $USER --policy-arn $policy_arn
> done
~ $
~ $ # Delete inline policies
~ $ for policy_name in $(aws iam list-user-policies --user-name $USER --query 'PolicyNames[]' --output text); do
>   aws iam delete-user-policy --user-name $USER --policy-name $policy_name
> done
~ $
~ $ # Remove from groups
~ $ for group in $(aws iam list-groups-for-user --user-name $USER --query 'Groups[].GroupName' --output text); do
>   aws iam remove-user-from-group --user-name $USER --group-name $group
> done
~ $
~ $ # Delete access keys
~ $ for key in $(aws iam list-access-keys --user-name $USER --query 'AccessKeyMetadata[].AccessKeyId' --output text); do
>   aws iam delete-access-key --user-name $USER --access-key-id $key
> done
~ $
~ $ # Delete signing certificates
~ $ for cert in $(aws iam list-signing-certificates --user-name $USER --query 'Certificates[].CertificateId' --output text); do
>   aws iam delete-signing-certificate --user-name $USER --certificate-id $cert
> done
~ $
~ $ # Delete SSH public keys
~ $ for ssh
>
Feedback
```

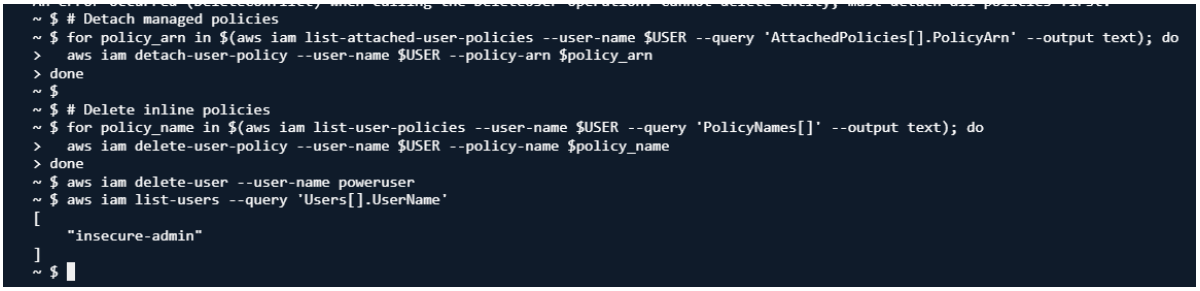
We open Trusted Advisor to review best practice checks for cost, security, and fault tolerance. Security checks show gaps to be addressed.

39. Trusted Advisor Security Findings

```
aws | Search [Alt+S]

CloudShell

us-east-1 +

> done
~ $
~ $ # Delete access keys
~ $ for key in $(aws iam list-access-keys --user-name $USER --query 'AccessKeyMetadata[].AccessKeyId' --output text); do
>   aws iam delete-access-key --user-name $USER --access-key-id $key
> done
~ $
~ $ # Delete signing certificates
~ $ for cert in $(aws iam list-signing-certificates --user-name $USER --query 'Certificates[].CertificateId' --output text); do
>   aws iam delete-signing-certificate --user-name $USER --certificate-id $cert
> done
~ $
~ $ # Delete SSH public keys
~ $ for ssh
> ^C
~ $ USER="poweruser"
~ $ # Delete SSH public keys
~ $ for ssh in $(aws iam list-ssh-public-keys --user-name $USER --query 'SSHPublicKeys[].SSHPublicKeyId' --output text); do
>   aws iam delete-ssh-public-key --user-name $USER --ssh-public-key-id $ssh
> done
~ $
~ $ # Delete login profile
~ $ aws iam delete-login-profile --user-name $USER 2>/dev/null
~ $
~ $ # Delete service-specific credentials
~ $ for svc in $(aws iam list-service-specific-credentials --user-name $USER --query 'ServiceSpecificCredentials[].ServiceSpecificCredentialId' --output text); do
>   aws iam delete-service-specific-credential --user-name $USER --service-specific-credential-id $svc
> done
~ $
~ $ # Delete MFA devices
~ $ for mfa in $(aws iam list-mfa-devices --user-name $USER --query 'MFADevices[].SerialNumber' --output text); do
>   aws iam deactivate-mfa-device --user-name $USER --serial-number $mfa
>   aws iam delete-virtual-mfa-device --serial-number $mfa
> done
~ $
~ $ # Finally, delete the user
~ $ aws iam delete-user --user-name $USER

An error occurred (DeleteConflict) when calling the DeleteUser operation: Cannot delete entity, must detach all policies first.
```

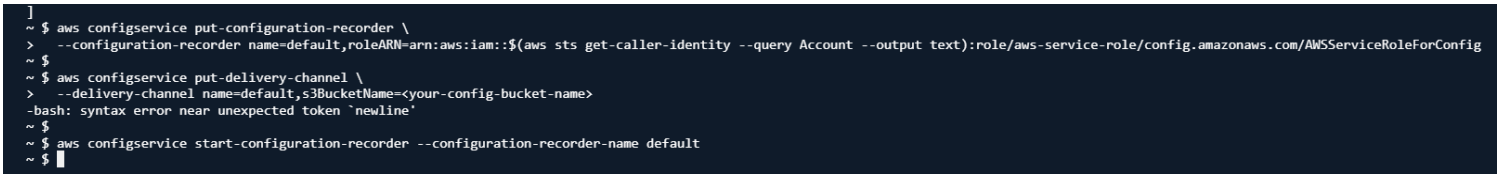
Trusted Advisor identifies exposed ports, weak password policies, and lack of MFA. These are quick wins to harden the environment.

40. Final Dashboard: CSPM Visibility Achieved



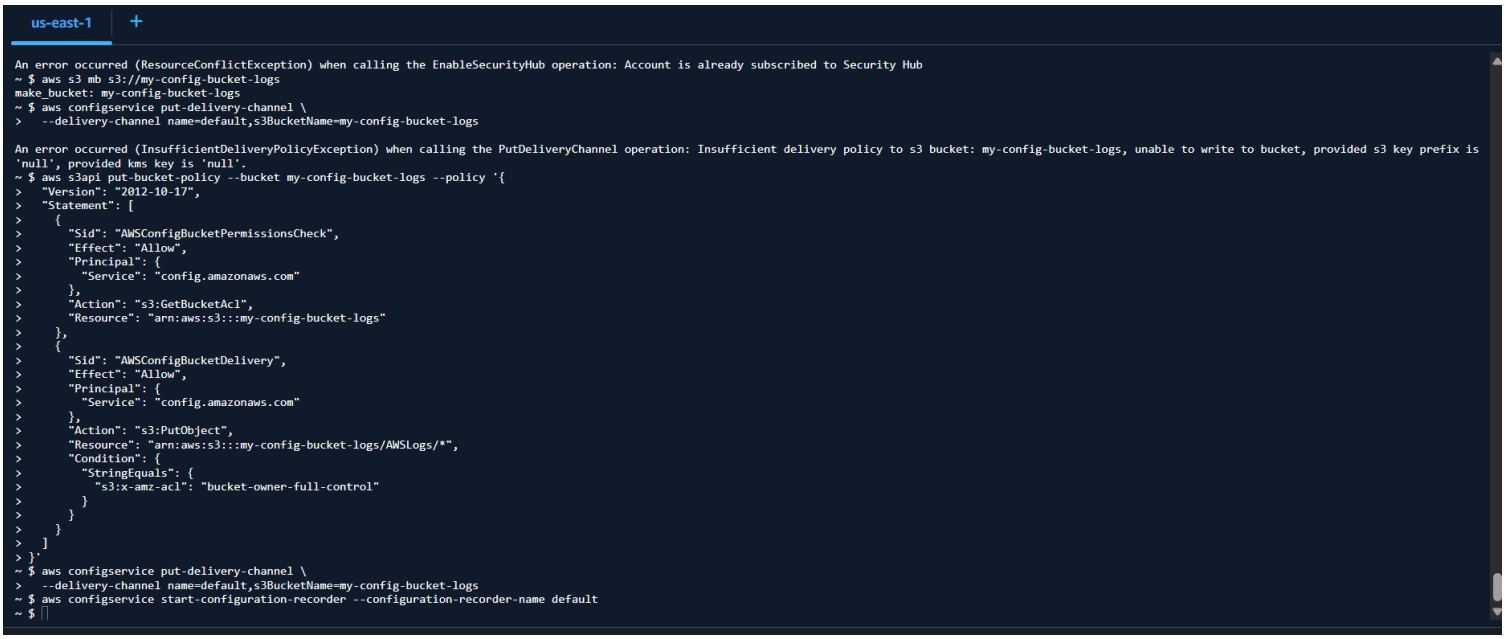
We conclude the CSPM assessment with a consolidated view across Config, Security Hub, GuardDuty, and Trusted Advisor. Risks are now clearly documented.

41. Configuration Recorder Setup Failure



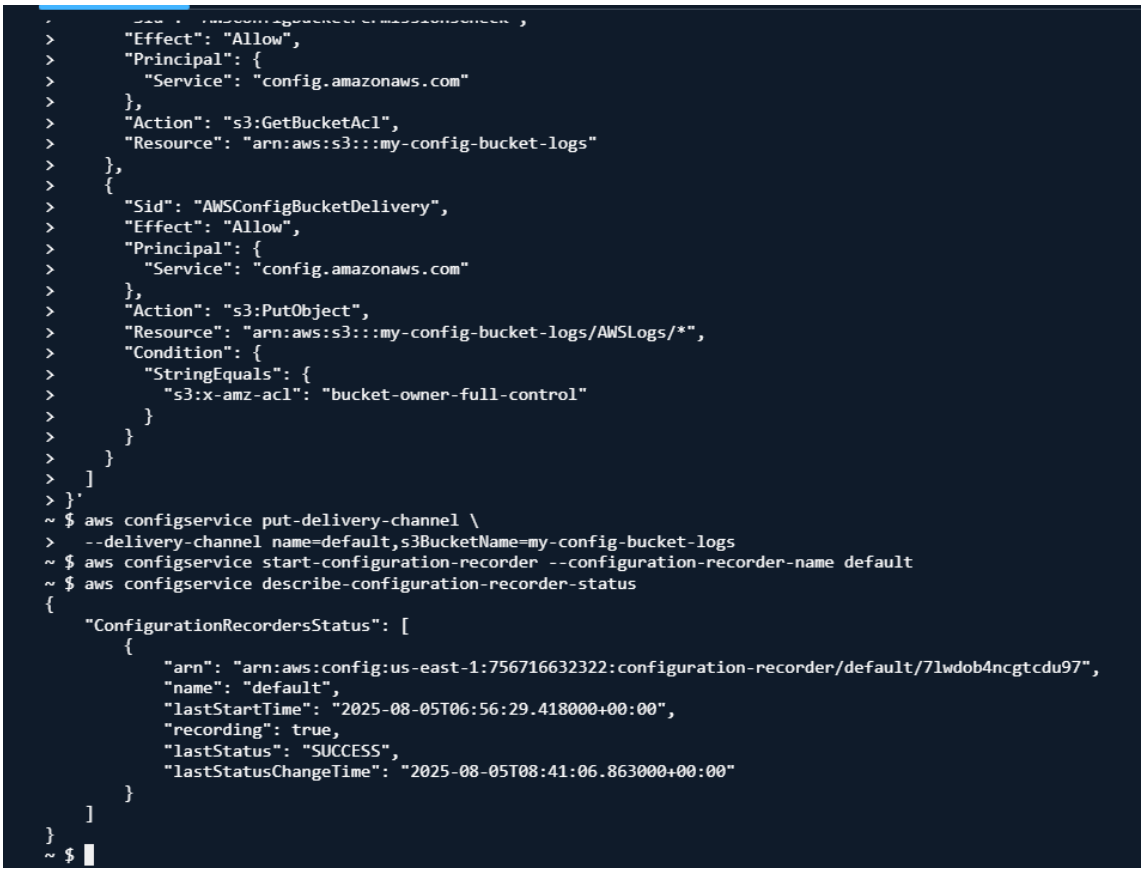
This attempt to set up the configuration recorder and delivery channel failed due to a syntax error and a missing bucket policy. Notice the syntax error message near the newline character and the lack of permission for the delivery channel to access the S3 bucket.

42. Fixing S3 Bucket Policy for AWS Config Delivery



Here, a proper bucket policy is added to the `my-config-bucket-logs` S3 bucket to allow AWS Config to deliver logs. This includes granting permissions to `config.amazonaws.com` for both `s3:GetBucketAcl` and `s3:PutObject` actions, with conditions ensuring full control by the bucket owner. The delivery channel command is then re-run successfully.

43. Configuration Recorder Running



Final verification of the configuration recorder shows it is successfully recording. The status shows `recording: true` and `lastStatus: SUCCESS`, confirming that AWS Config is now operational.

This AWS Cloud Security Posture Assessment (CSPM) demonstration provided a full walkthrough of identifying, validating, and remediating common misconfigurations in an AWS environment. It showcased how to leverage native AWS security tools — including AWS Config, CloudTrail, Security Hub, GuardDuty, and Trusted Advisor — to gain visibility, enforce compliance frameworks like CIS, and reduce security risk.

By following this methodology, organisations can:

- Establish continuous monitoring and compliance tracking.
- Detect high-risk misconfigurations early and remediate them efficiently.
- Improve readiness for audits against standards such as CIS, NIST, and SOC 2.
- Maintain a strong, resilient security posture in the cloud.

Ongoing Practice
Regularly reviewing configuration baselines, validating access controls, and integrating findings into security operations ensures sustained cloud compliance and resilience. Combining AWS native security services with a structured remediation process creates an environment where security and compliance are continuously maintained, not just assessed periodically.