



Secure AWS Multi-Account Baseline

*Terraform +
Policy-as-Code*

*Portfolio
by Amina Jiyu An
[@GitHub](#)*

Executive Summary

Enterprise AWS Secure Baseline (Terraform + Policy-as-Code)

“This is not theory. Every slide is a proof I built, validated, and enforced in AWS.”

This portfolio demonstrates how I designed and enforced a secure AWS environment at **enterprise scale**, combining preventive, detective, and governance controls. Every component is mapped to **international and regional compliance frameworks** (ISO/IEC 27001, Saudi NCA ECC, UAE NESAS), proving awareness of both global standards and local regulatory requirements.

✓ Multi-account Governance

Multi-account governance with AWS Orgs & SCPs) →

ISO 27001 A.5.1, NCA GOV-02, NESAS GOV-01.

✓ Centralized Logging Encryption

CloudTrail + S3/KMS logs
→
ISO 27001 A.12.4/8.15 |
NCA LGM-02 | NESAS
MON-01

✓ Compliance Mapping

AWS Config Conformance Packs →

ISO 27001 A.12.1/5.14 |
NCA CC-06 | NESAS
AUD-02

✓ Threat detection & CSPM

GuardDuty + Security Hub →

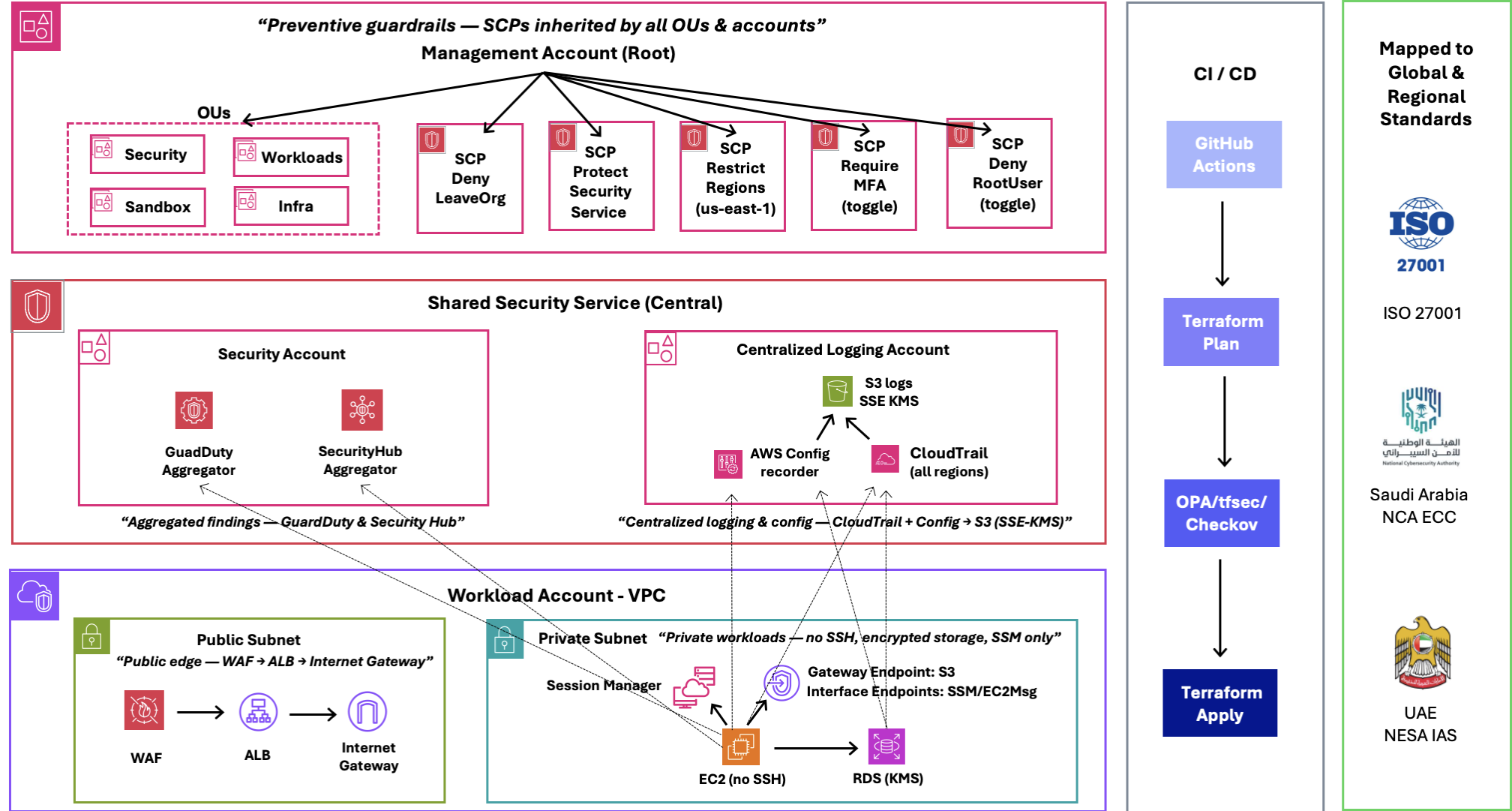
ISO 27001 A.12.6/8.8 |
NCA D5.5 | NESAS
MON-05

✓ Policy-as-Code (OPA,tfsec,Checkov)

Enforce encryption & IAM boundaries →

ISO 27001 A.14.2/8.28 |
NCA D3.2 | NESAS DEV-01

Architecture Diagram



Preventive = SCPs • Detective = CloudTrail/Config/GD/SecHub • Foundational = VPC+Encryption+SSM • Governance = Terraform+OPA

Validated against
 ISO 27001, Saudi NCA ECC, UAE NESAS IAS controls

Figure 1: Secure AWS multi-account architecture with governance, logging, and workload layers

Step1: State Backend

What this proves

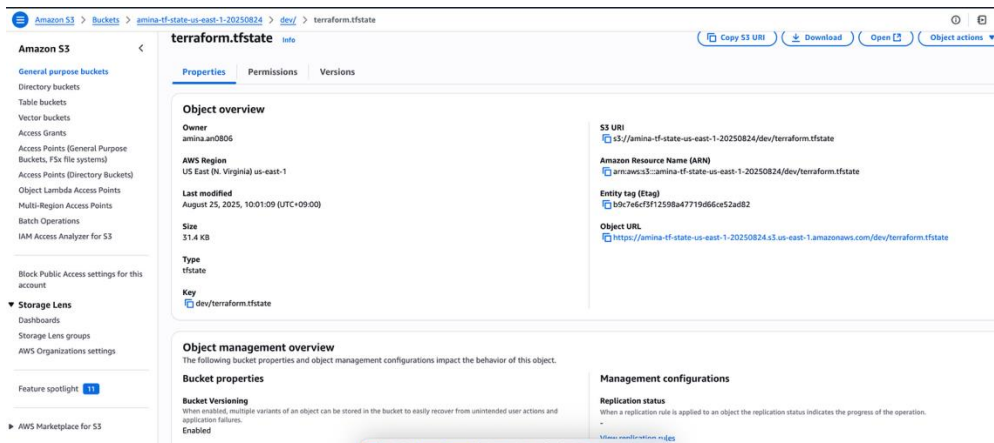
- Secure Terraform state management across accounts.
- Encryption (SSE-KMS) protects state confidentiality.
- DynamoDB locking prevents concurrent writes / corruption.

Controls

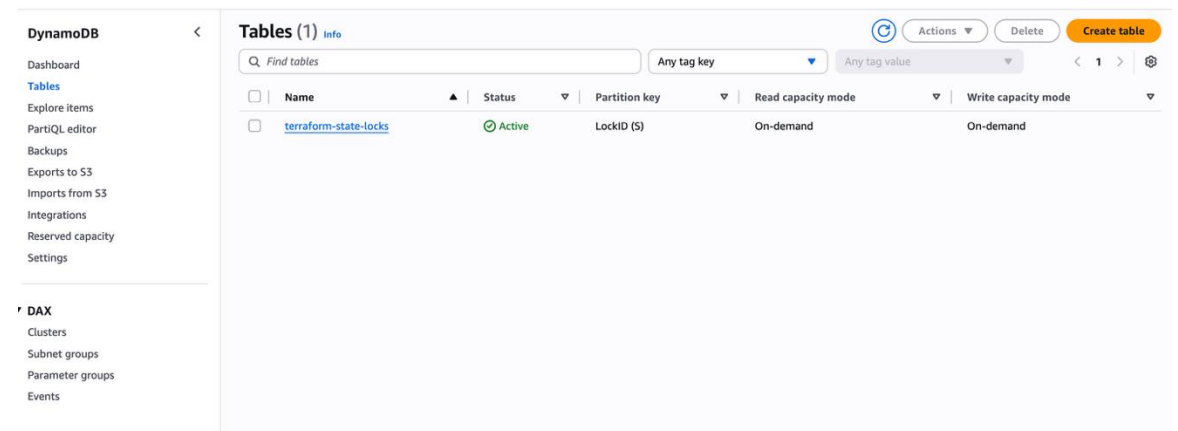
- ISO 27001: A.8.20, A.8.23, A.8.16 → 2022: 8.24, 5.23, 5.15

Proofs / Screenshots

S3 bucket (SSE-KMS enabled)



DynamoDB state locking



Note: Ensures **tamper-resistant, segregated state** → critical for enterprise IaC.

Step2: Centralized Logging

What this proves

- Enterprise-wide **visibility** into all AWS activity.
- CloudTrail & AWS Config logs are **centralized, encrypted, immutable**.
- S3 bucket with **KMS CMK + versioning** → no accidental/intentional log deletion.

Controls

- ISO 27001: A.12.4 → 2022: 8.15
- Saudi NCA: D1 Logging & CC-06 Compliance
- UAE NESa: MON-01

Proofs / Screenshots

Log bucket encryption (SSE-KMS)

The screenshot shows the Amazon S3 console interface. On the left, a sidebar lists navigation options under 'Amazon S3', including 'General purpose buckets', 'Directory buckets', 'Table buckets', 'Vector buckets', 'Access Grants', 'Access Points (General Purpose Buckets, FSx file systems)', 'Access Points (Directory Buckets)', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', and 'IAM Access Analyzer for S3'. The main content area is titled 'Default encryption' with an 'Info' link and an 'Edit' button. It states: 'Server-side encryption is automatically applied to new objects stored in this bucket.' Below this, the 'Encryption type' is 'Server-side encryption with AWS Key Management Service keys (SSE-KMS)'. The 'Encryption key ARN' is displayed as 'arn:aws:kms:us-east-1:958006149724:key/mrk-27d3409cf7c04b4ea998f16c7ae654a0'. The 'Bucket Key' section indicates it is 'Enabled'. At the bottom, there are 'Intelligent-Tiering Archive configurations (0)' with buttons for 'View details', 'Edit', 'Delete', and 'Create configuration'. A search bar for configurations is also present.

CloudTrail Logs in S3

The screenshot shows the 'Objects (21)' view in the Amazon S3 console. At the top, there are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', and 'Create folder'. An 'Upload' button is also visible. Below the buttons, a search bar says 'Find objects by prefix' and a toggle for 'Show versions' is set to 'Off'. A table lists the objects with columns for checkboxes, Name, Type, Last modified, Size, and Storage class. Three objects are visible, all of type 'gz' and 'Standard' storage class. The first object is '958006149724_CloudTrail_us-east-1_20250825T0105Z_hvMrzlypfEgw0KR.json.gz' (2.1 KB, modified Aug 25, 2025, 10:07:27 UTC+09:00). The second is '958006149724_CloudTrail_us-east-1_20250825T0105Z_kmnUHK4lmQeRaj.json.gz' (6.0 KB, modified Aug 25, 2025, 10:06:53 UTC+09:00). The third is '958006149724_CloudTrail_us-east-1_20250825T0105Z_LsIX' (893.0 B, modified Aug 25, 2025, 10:07:40 UTC+09:00).

Note: Logging is the foundation for monitoring & audit evidence.

Step3: AWS Config & Conformance Packs

What this proves

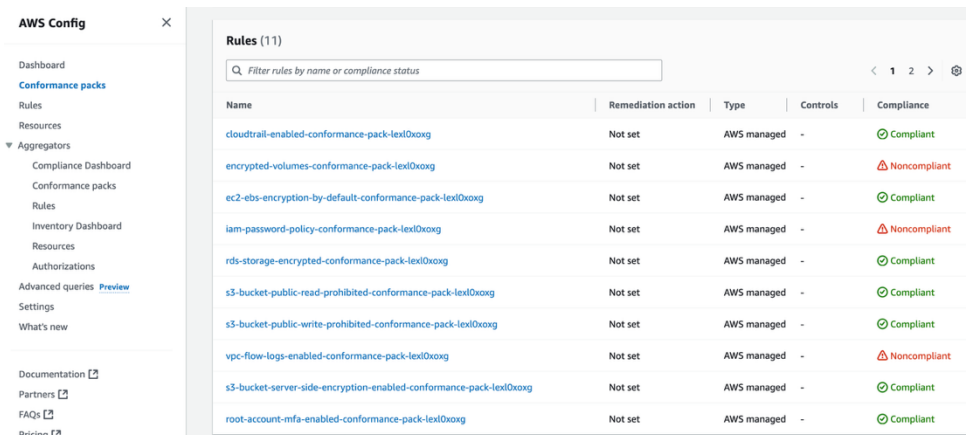
- **Detects misconfigurations** → flags non-compliance in near real-time.
- Conformance Pack with 11 security baseline rules (passwords, MFA, encryption, logs)

Controls

- ISO 27001: A.12.1, A.18.2.2 → 2022: 5.14, 5.36
- Saudi NCA ECC: OAM-06 (config mgmt.)
- UAE NESIA IAS: Secure baseline, data protection, audit & accountability

Proofs / Screenshots

Config Rules Evaluations



Name	Remediation action	Type	Controls	Compliance
cloudtrail-enabled-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Compliant
encrypted-volumes-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Noncompliant
ec2-ebs-encryption-by-default-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Compliant
iam-password-policy-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Noncompliant
rd5-storage-encrypted-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Compliant
s3-bucket-public-read-prohibited-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Compliant
s3-bucket-public-write-prohibited-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Compliant
vpc-flow-logs-enabled-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Noncompliant
s3-bucket-server-side-encryption-enabled-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Compliant
root-account-mfa-enabled-conformance-pack-lexl0xoxg	Not set	AWS managed	-	Compliant

CLI Conformance Pack

```
Janes-MacBook-Pro:~ janeahn$ aws configservice describe-conformance-packs --conformance-pack-names starter-dev
{
  "ConformancePackDetails": [
    {
      "ConformancePackName": "starter-dev",
      "ConformancePackArn": "arn:aws:config:us-east-1:958006149724:conformance-pack/starter-dev/conformance-pack-lexl0xoxg",
      "ConformancePackId": "conformance-pack-lexl0xoxg",
      "DeliveryS3Bucket": "awsconfigconforms-baseline-958006149724-us-east-1",
      "DeliveryS3KeyPrefix": "artifacts",
      "ConformancePackInputParameters": [],
      "LastUpdateRequestedTime": "2025-08-26T21:27:59.086000+09:00"
    }
  ]
}
```

Note: Provides ongoing evidence for audits. Moves from reactive audits → proactive continuous compliance

Step4: Security Hub & GuardDuty

What this proves

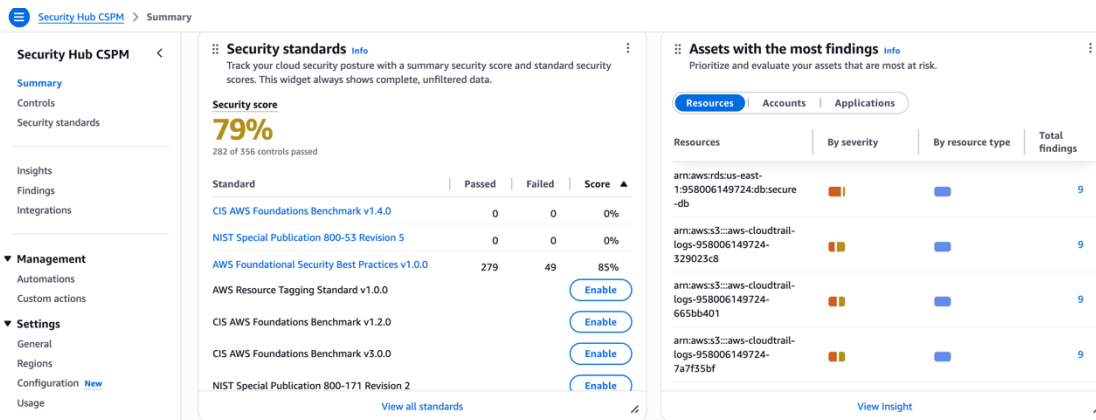
- Unified **threat detection + compliance aggregation**.
- Security Hub consolidates findings (CIS, PCI DSS).
- GuardDuty detects **anomalous network and account behavior**.

Controls

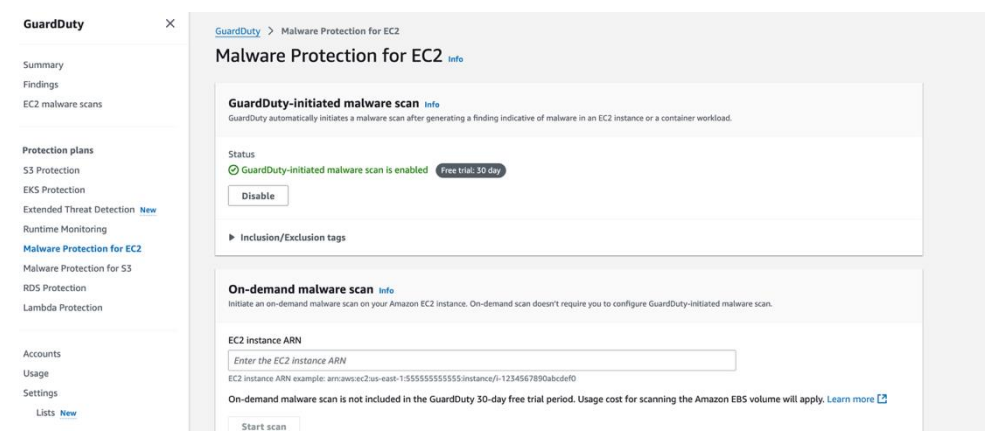
- ISO 27001: A.12.6, A.16.1 → 2022: 8.8, 5.25
- Saudi NCA: D5.5 Threat Detection

Proofs / Screenshots

Security Hub Summary



GuardDuty Detector ON



Note: Provides central view of risk posture across all accounts

Step5: Policy-as-Code (OPA, tfsec, Checkov)

What this proves

- **Automated governance** before provisioning.
- Prevents deployment of insecure resources (unencrypted S3, missing MFA, etc.).
- CI/CD gate → code must pass tfsec, Checkov, OPA rules **before** apply.

Controls

- ISO 27001: A.14.2, A.12.1.2, A.18.2.3 → 2022: 8.28, 5.14, 5.35
- CIS AWS Foundations: enforced via Hub CIS subscription

Proofs / Screenshots

✗ OPA eval fail (GuardDuty missing)

```
Janes-MacBook-Pro:tf-aws-secure-baseline janeahn$ cat > plan-no-guardduty.json <<'EOF'
> {
>   "resource_changes": []
> }
> EOF
Janes-MacBook-Pro:tf-aws-secure-baseline janeahn$ opa eval -d policies-as-code/opa -i plan-no-guardduty.json 'data.terraform.security.m
tty
{
  "GuardDuty is not being enabled in this plan (missing aws_guardduty_detector).",
```

✓ OPA eval pass (all checks)

```
Janes-MacBook-Pro:tf-aws-secure-baseline janeahn$ opa eval -d policies-as-code/opa -i plan-pass.json 'data.terraform.security.result' -f pretty
{
  "count": 0,
  "messages": [],
  "passed": true
}
Janes-MacBook-Pro:tf-aws-secure-baseline janeahn$
```

✓ OPA unit tests pass

```
Janes-MacBook-Pro:tf-aws-secure-baseline janeahn$ opa test policies-as-code/opa -v
policies-as-code/opa/policies_test.rego:
data.terraform.security.test_guardduty_missing_detector_denies: PASS (2.876834ms)
data.terraform.security.test_iam_group_missing_boundary_denies: PASS (3.512292ms)
data.terraform.security.test_guardduty_disabled_denies: PASS (3.549917ms)
data.terraform.security.test_securityhub_both_missing_denies: PASS (3.554584ms)
data.terraform.security.test_securityhub_account_only_denies_subscription: PASS (3.612166ms)
data.terraform.security.test_iam_role_missing_boundary_denies: PASS (3.591125ms)
data.terraform.security.test_iam_user_missing_boundary_denies: PASS (3.576917ms)
data.terraform.security.test_s3_requires_sse_denies: PASS (626.75µs)
data.terraform.security.test_s3_with_sse_allows: PASS (4.415333ms)
data.terraform.security.test_guardduty_enabled_allows: PASS (1.988041ms)
-----
PASS: 10/10
```

Note: Shifts compliance left → security embedded in development pipeline.

Step6: Organizations & SCPs

What this proves

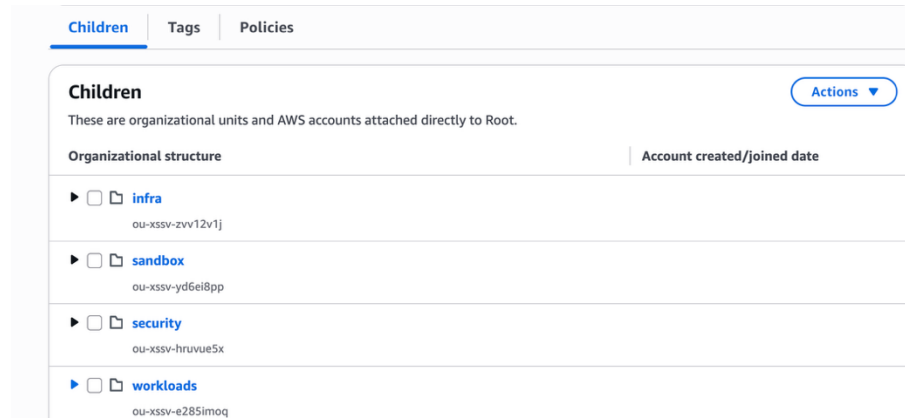
- **Preventive guardrails** enforced Org-wide.
- **SCPs restrict dangerous actions:** Deny leaving Org , Protect CloudTrail/Config/SecHub/GD , Restrict regions (only us-east-1), Require MFA for IAM writes, Deny root user access (toggle)

Controls

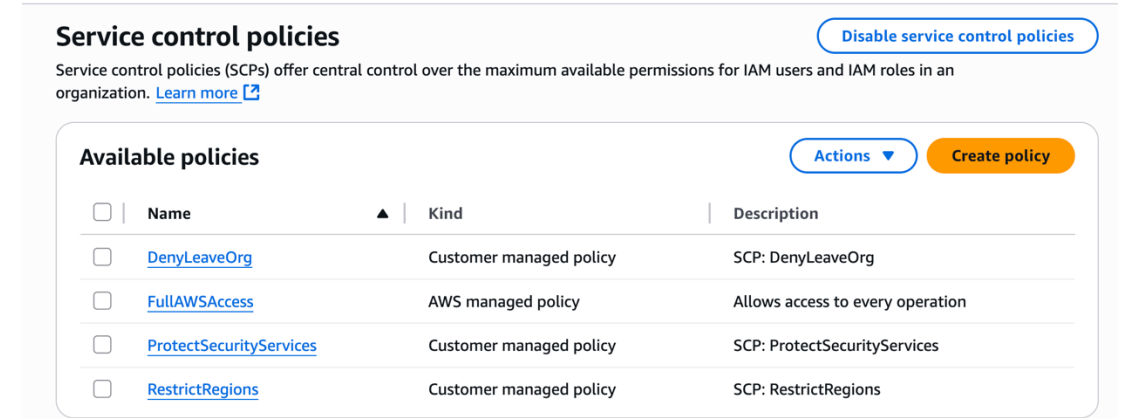
- ISO 27001: A.5.1.1, A.9.2.3, A.9.2.1 → 2022: 5.1, 5.18, 5.17
- Saudi NCA: GOV-02
- UAE NESI: GOV-01

Proofs / Screenshots

Org OUs (security, infra, workloads, sandbox)



Root with SCPs Attached



Note: These guardrails prevent violations at source — stronger than detective controls.

Compliance Mapping

Mapping of AWS security controls → ISO/IEC 27001, Saudi NCA ECC, and UAE NESAIAS frameworks, demonstrating awareness of both global standards and regional compliance mandates.

Step	Implementation Example	ISO/IEC 27001 (2013→2022)	CIS AWS Foundations	Saudi NCA ECC	UAE NESAIAS
1 — State Backend	S3 backend SSE-KMS, DynamoDB lock	A.8.20 → 8.24 (crypto), A.8.16 → 5.15 (access control)	2.2 log encryption, 2.1.1 public access blocked	—	—
2 — Centralized Logging	CloudTrail, CloudWatch, KMS	A.12.4 → 8.15 (logging), A.8.20 → 8.24	2.1 all regions, 2.2 validation, 2.3 CMKs	D1/D2 logging & monitoring	Logging & monitoring
3 — Config & Conformance	Config rules, Conformance packs	A.12.1 → 5.14, A.18.2.2 → 5.36, A.12.7 → 5.35	2.5 Config enabled, 2.6 all resources	CC-06 compliance checks	Compliance & audit governance
4 — Security Hub & GuardDuty	Threat detection, incident dashboard	A.12.4 → 8.15, A.12.6 → 8.8, A.16.1 → 5.25	3.1 GuardDuty, 3.2 Security Hub	D5.5 threat detection, CC-06	Threat & vulnerability management, security monitoring
5 — OPA Policy-as-Code	Terraform plan eval, CI/CD enforcement	A.12.6 → 8.8, A.12.4 → 8.15, A.18.2.2 → 5.36, A.9.2.3 → 5.18	1.1 MFA, 2.x log checks, 3.x GuardDuty/SecHub	D3.2 secure by design, D5.3 IAM	Secure development lifecycle, automated compliance
6 — Organizations & SCPs	DenyLeaveOrg, Protect Security Services, RestrictRegions	A.5.1.1 → 5.1, A.12.4 → 8.15, A.9.1.2 → 5.12, A.9.2.3 → 5.18	1.1 MFA, 1.5 IAM, 1.6 Root disabled, 2.1 CloudTrail	D5.2 IAM, D5.5 GuardDuty, D1/D2 logging, CC-06	Governance, access control, continuous monitoring

CI/CD Enforcement with Policy-as-Code

What this proves

- GitHub Actions pipeline runs **security checks automatically**.
- Every pull request triggers **tfsec, Checkov, OPA** before merge.
- Pipeline ensures “**no code is applied without passing security gates.**”

Controls

- ISO 27001: **A.14.2 Secure coding, A.12.1.2 Change management, A.18.2.3 Technical compliance review** → 2022: **8.28, 5.14, 5.35**
- NCA: **DEV-01 Secure Dev, CC-06 Compliance checks**
- NESA: **DEV-01, AUD-02**

Proofs / Screenshots

GitHub Actions YAML workflow

```
name: terraform-security-checks

on:
  pull_request:
    paths:
      - "**/*.tf"
      - "policies-as-code/**"
      - ".github/workflows/plan.yml"
  workflow_dispatch:

jobs:
  security_checks:
    runs-on: ubuntu-latest
    env:
      # Default guess; we'll auto-correct this below if the folder doesn't exist
      WORK_DIR: envs/dev
      OPA_DIR: policies-as-code/opa
```

CI Badge Green



Enterprise AWS Secure Baseline (Terraform + PaC)

This project demonstrates **how to design and enforce a secure AWS environment at enterprise scale**. It includes:

- Multi-account setup with AWS Organizations & Service Control Policies
- Centralized logging (CloudTrail, CloudWatch, S3 + KMS)
- AWS Config Conformance Packs for compliance monitoring
- Security Hub & GuardDuty as Cloud Security Posture Management (CSPM) tools
- Policy-as-Code (OPA/Rego) to enforce encryption, IAM boundaries, and security service activation

📁 Compliance Mapping: **ISO/IEC 27001 Annex A (2013 & 2022), Saudi NCA ECC, UAE NESA IAS**

ISO/IEC 27001 Annex A — Control Mapping (2013 → 2022)

Key takeaway: IaC merges are blocked unless security & compliance tests pass → demonstrates real DevSecOps maturity and ability to operationalize cloud security pipelines.

Conclusion

“Every slide is a proof I built, validated, and enforced in AWS.”

- ✓ **Compliance → ISO 27001, NCA ECC, NESA IAS**
- ✓ **Security → SCPs, Logging, GuardDuty, Security Hub**
- ✓ **DevSecOps → Policy-as-Code in CI/CD pipelines**

Amina Jiyu An

Cloud Security & Compliance Engineer/Architect