## Project: Secure the Cloud – Phase 1: Identity Governance

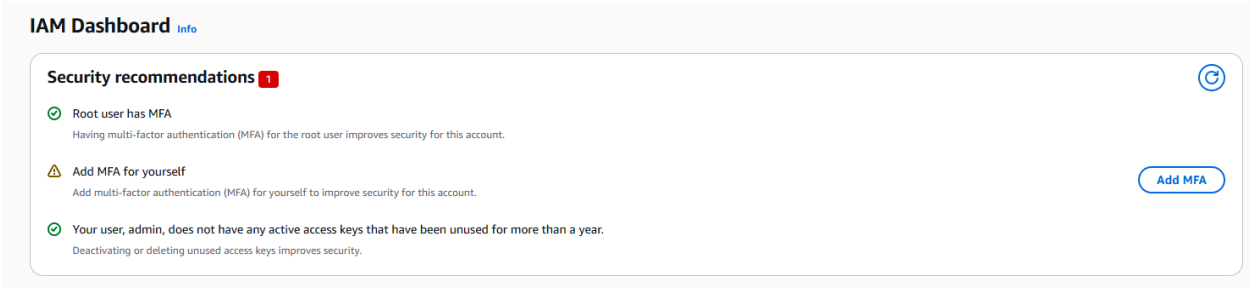
**Objective:** To architect a production-ready AWS IAM environment. This lab demonstrates the transition from a high-risk "Root" account to a **Least Privilege** model using **Role-Based Access Control (RBAC)**.

---

### Step 1: Root Account Hardening & Defense-in-Depth

**Action:** Initialized the environment by securing the Root user.

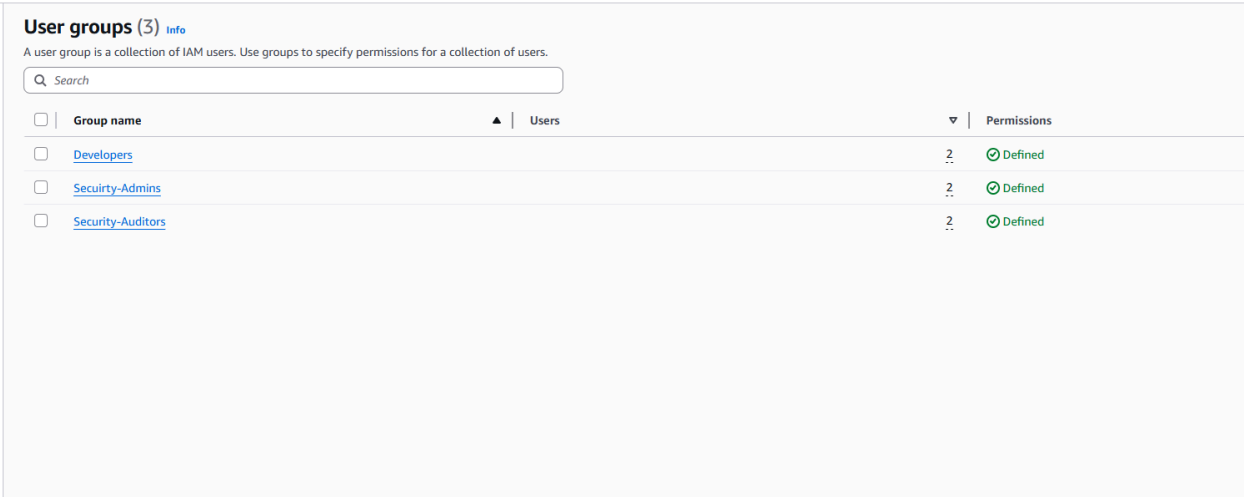
- **Why it's Critical:** Using the Root account for daily operations is a critical vulnerability; a single credential leak could lead to total account takeover or "credential stealing".
- **Implementation:** Enabled **Multi-Factor Authentication (MFA)** on the Root account to provide a secondary layer of protection.




### Step 2: Engineering the RBAC Hierarchy (Groups)

**Action:** Developed a group-based permission structure to manage access by job function.

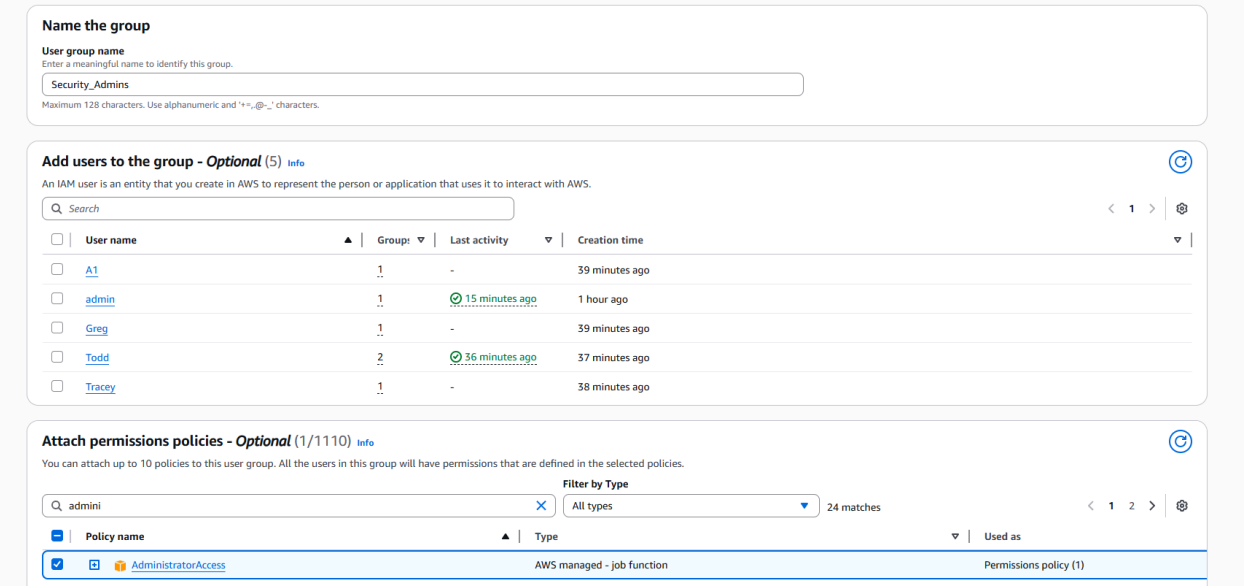
- **Group 1: Security-Admins:** Assigned `AdministratorAccess` for high-level environment management.
- **Group 2: Security-Auditors:** Assigned `SecurityAudit` permissions for read-only monitoring and log analysis.
- **Group 3: Developers:** Assigned `AmazonS3ReadOnlyAccess` to allow resource viewing without modification.




### Step 3: Identity Provisioning (Users)

**Action:** Provisioned 5 distinct identities and mapped them to their functional groups.

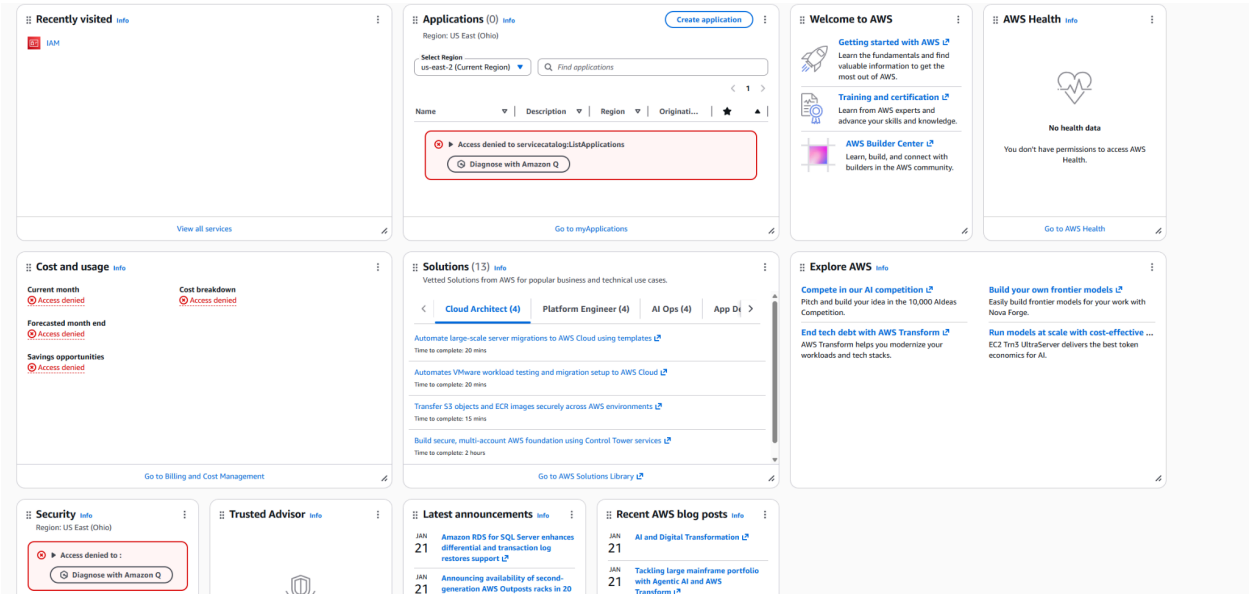
- **Admins:** `admin`, `Todd` (temporarily restricted for testing).
- **Auditors:** `A1` and `Todd`.
- **Developers:** `Greg` and `Tracey`.




### Step 4: Security Validation (The "Access Denied" Test)

**Action:** Logged in as the `Todd` account (Auditor role) to verify security boundaries.

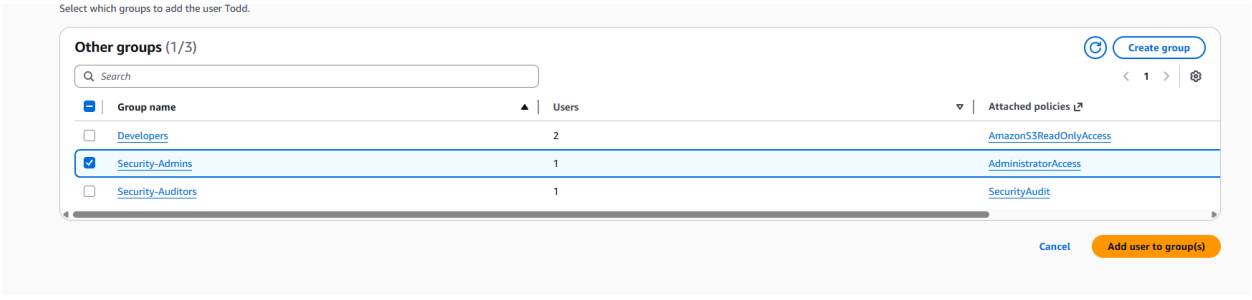
- **The Test:** Attempted to perform administrative actions (such as viewing billing or changing security settings) while restricted to the Auditor group.
- **The Result:** The system returned an **"Access Denied"** message, confirming that the **Principle of Least Privilege** is active and the "Blast Radius" of a standard user is limited.




### Step 5: Operational Flexibility (Escalation of Privileges)

**Action:** Demonstrated how an administrator can safely escalate a user's permissions by moving them between groups.

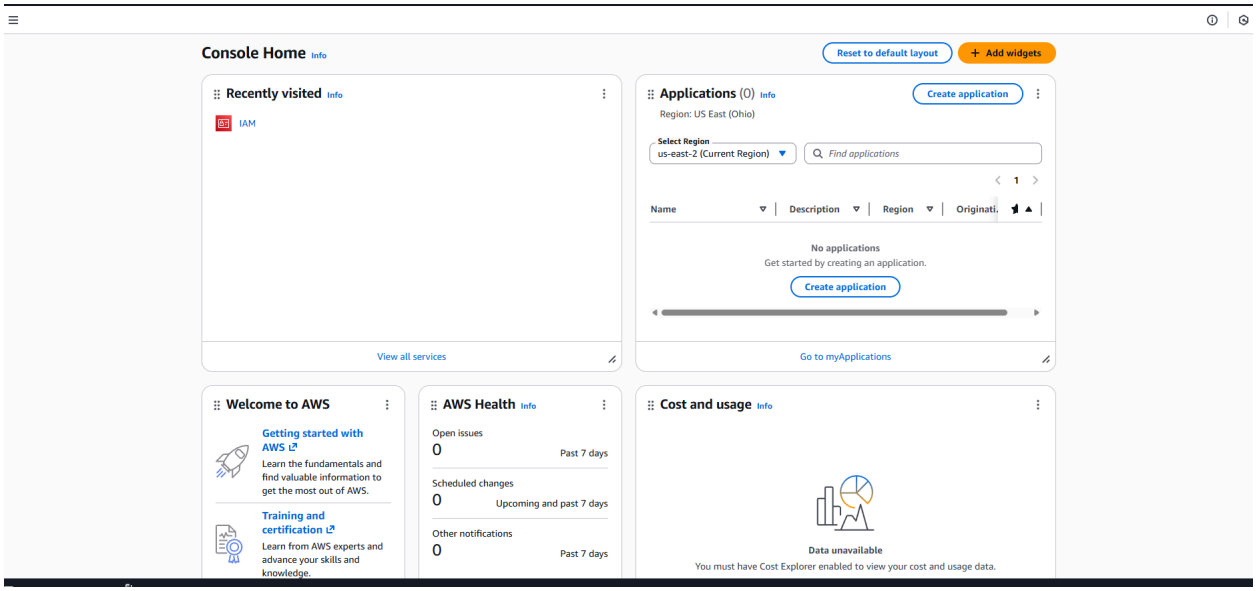
- **Procedure:** Moved `Todd` from the `Security-Auditors` group to the `Security-Admins` group.
- **Validation:** Refreshed the session to confirm the user now has full access to previously restricted data.




## 6. Privilege Escalation & Operational Lifecycle

**Objective:** To demonstrate the secure management of user permissions through group membership transitions.

- **Scenario**: The user `Todd` requires temporary administrative access to perform a high-level system configuration.
- **Procedure**:
    1. Navigated to the **IAM Users** dashboard and selected the `Todd` identity.
    2. Accessed the **Permissions** tab and selected **"Add user to group"**.
    3. Selected the `Security-Admins` group and confirmed the addition.
- **Validation**: Upon re-authentication, the `Todd` account successfully accessed the "Security" and "Cost and Usage" data that was previously restricted.




---

## 7. Final Project Summary

By architecting this environment, I have achieved two major security milestones:

- **Root Account De-escalation**: Eliminated the daily risk of Root account compromise by enforcing **MFA** and transitioning to standard administrative users.
- **Scalable Governance**: Created a structured system of **Users and Groups** that allows for rapid onboarding and offboarding without modifying individual permission sets.

---

## 8.Analyst Closing Note

This concludes **Phase 1: Identity Governance**. The environment is now hardened, audited, and ready for **Phase 2**, where I will implement automated security monitoring to detect resource misconfigurations in real-time.

**[PHOTO 6: Final AWS Console Dashboard showing all services accessible to the new Admin user]**

---

## Final Review & Takeaways

- **Principal of Least Privilege:** Proven by the successful blocking of unauthorized actions by the auditor account.
- **Blast Radius Reduction:** By segmenting developers and auditors, the compromise of a single account no longer poses a threat to the entire infrastructure.
- **NIST CSF Alignment:** This project implements the **Identity Management and Access Control (PR.AC)** functions of the NIST framework.