Cloud Security with AWS IAM

Author: Rishi Puranik

R by Rishi Puranik

Introduction

In modern cloud security, identity and access management (IAM) plays a critical role in ensuring that only authorized users can interact with cloud resources. AWS IAM provides a structured way to manage permissions, enforce security policies, and control resource access efficiently. In this project, I configured AWS IAM to implement role-based access control (RBAC) by restricting access to cloud environments based on user roles. Specifically, I created an IAM setup that granted an "intern" account access to a development environment while restricting access to the production environment.

This project demonstrates best practices in least privilege enforcement, policy-based access control, and secure cloud resource management.

AWS IAM Implementation

IAM User Groups & Role-Based Access

To streamline access control, I created an IAM user group and assigned the intern account to it. This ensured that permissions could be managed at the group level rather than individually, maintaining consistency and scalability.

- Intern User Group → Granted access to development resources only.
- Production Access Restriction → Prevented unauthorized changes to critical systems.

This approach follows industry best practices, as managing access via groups simplifies policy enforcement while reducing the risk of human error in permission management.

IAM Policies & JSON Configuration

AWS IAM policies define what actions are allowed or denied. To enforce controlled access, I wrote a JSON policy that:

- ✅ Allowed unrestricted access to EC2 instances tagged as development.
- ❌ Denied any modification or deletion of tags, preventing privilege escalation.
- ❌ Restricted access to production instances, ensuring operational integrity.

Example JSON snippet:

```
{
 "Effect": "Allow",
 "Action": "ec2:DescribeInstances",
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "ec2:ResourceTag/Env": "Development"
  }
 }
}
```

This policy ensures that users can interact with development instances while protecting production resources from unauthorized modifications.

Using Tags for Access Management

AWS tags provide a simple way to categorize and manage resources. In this project, I used an "Env" tag with values:

- Development → Allowed full interaction by the intern user.
- Production → Completely restricted.

This tagging approach enhances security by ensuring that permissions are dynamically applied based on the resource's classification.

Testing IAM Policies

To verify the correctness of the IAM policy, I tested its enforcement:

- ✔️ Stopping Development Instance: Allowed as expected.
- ❌ Stopping Production Instance: Blocked due to policy restrictions.

This confirmed that the principle of least privilege (PoLP) was properly enforced, preventing unauthorized actions.

Additional Security Enhancements

Account Alias for Simplified Login

To improve user experience, I configured an AWS account alias for easy login. Instead of using a random account ID, users can now sign in using:

https://nextwork-alias-rishipuranik.signin.aws.amazon.com/console

IAM Security Considerations

While this setup follows security best practices, additional measures could be implemented:

- Multi-Factor Authentication (MFA): Further securing IAM users against unauthorized access.
- IAM Access Analyzer: Detecting overly permissive policies to reduce attack surface.
- Logging & Monitoring: Using AWS CloudTrail to track IAM activity and audit access attempts.

Key Takeaways & Lessons Learned

- IAM policies provide fine-grained control over cloud security when designed correctly.
- User groups & role-based access simplify permission management and reduce errors.
- Tag-based access control is an effective way to enforce environment-specific restrictions.
- Testing policies before deployment is crucial to prevent security misconfigurations.

Time to Implement: Less than 1 hour

The efficiency of AWS IAM's structured policy framework allowed for rapid deployment and testing of this setup.

Skills Used & Developed

- ✅ AWS IAM Role & Policy Design
- ✅ JSON-based IAM Policy Writing
- ✅ Least Privilege Enforcement
- ✅ AWS EC2 Access Management
- ✅ Cloud Security Best Practices

Final Thoughts

This project provided hands-on experience with AWS IAM, reinforcing security principles that are essential in real-world cloud environments. With additional features such as MFA, logging, and anomaly detection, this setup could be extended for enterprise-grade security.