Splunk Component Rehydration with AWS Lambda

As part of my responsibilities during the security engineering internship, I developed and deployed a Lambda-based solution to automate the rehydration of Splunk EC2 components. This initiative was critical for meeting strict financial compliance requirements while maintaining continuous availability and correct configuration of Splunk infrastructure.

I designed and implemented key AWS components to support this automation, including:

- **Lambda Function** – Programmatically handled teardown and rebuild of EC2 instances tied to Splunk roles such as Search Heads, Indexers, and Forwarders.
- **16 EventBridge Rules** – Created distinct EventBridge rules to monitor EC2 lifecycle events (start, stop, terminate, launch) across various Splunk environments, triggering the Lambda function dynamically.
- **IAM Roles & Policies** – Built fine-grained IAM roles and permissions to grant the Lambda function secure access to EC2, CloudWatch, and tagging services while maintaining least privilege.
- **CloudWatch Logging** – Enabled detailed logging of function execution for real-time monitoring, operational insight, and debugging.
- **Tagging & Auditing Module** – Applied standardized and traceable tags during instance rebuilds for visibility, resource tracking, and audit compliance.
- **Health Validation Checks** – Performed post-launch validations to confirm Splunk services (e.g., search head clustering or indexer readiness) were operational and properly connected.
- **Compliance Automation** – Ensured rehydration activities aligned with financial regulatory policies around uptime, resource configuration, and change traceability.

Through this project, I gained in-depth experience with AWS Lambda, EC2 automation, event-driven architectures, and compliance-aligned DevOps workflows. The solution enhanced operational resilience, minimized manual intervention, and helped ensure the Splunk infrastructure remained consistently healthy and regulation-ready.