Observability Setup for Splunk

**Phase 1: Planning and Requirements Gathering**

**Step 1.1: Identify Key Objectives for Observability**

The objectives for observability should align with your business goals. Here are some common objectives:

* **Performance Optimization:** Monitor and optimize the performance of AWS infrastructure and applications.
  + Example: Ensure minimal latency for APIs or applications.
* **Cost Optimization:** Track and analyze AWS resource utilization to reduce waste and control costs.
  + Example: Identify underused EC2 instances or unoptimized Lambda functions.
* **Fault Detection and Resolution:** Detect, diagnose, and resolve system issues before they impact users.
  + Example: Proactively monitor for high error rates in critical services.
* **Security and Compliance:** Ensure security measures are in place and comply with regulatory requirements.
  + Example: Monitor CloudTrail logs for unauthorized activities.

**Step 1.2: List AWS Services to Monitor**

Identify the AWS services relevant to your infrastructure. Common services include:

* **Compute:**
  + EC2 (virtual servers)
  + Lambda (serverless compute)
  + ECS/EKS (container orchestration)
* **Storage:**
  + S3 (object storage)
  + EBS (block storage)
  + EFS (shared file system)
* **Databases:**
  + RDS (relational databases)
  + DynamoDB (NoSQL database)
* **Networking:**
  + VPC (network isolation)
  + ELB (load balancers)
  + CloudFront (content delivery)
* **Monitoring and Logging:**
  + CloudWatch (metrics and logs)
  + CloudTrail (audit logs)

**Step 1.3: Define Key Performance Indicators (KPIs)**

KPIs depend on the services and objectives. Below are examples for common AWS services:

* **EC2 Instances:**
  + CPU Utilization
  + Memory Usage
  + Disk IO
  + Network In/Out
* **RDS Databases:**
  + CPU Utilization
  + Connections Count
  + Read/Write Latency
  + Backup Storage Size
* **Lambda Functions:**
  + Invocation Count
  + Average Duration
  + Error Count
  + Throttles
* **S3 Buckets:**
  + Total Storage Size
  + Request Count (GET, PUT, DELETE)
  + Error Rates (403, 404)
* **CloudFront:**
  + Cache Hit Ratio
  + Latency
  + HTTP 4xx/5xx Errors

**2. Assess Resources**

**Step 2.1: Splunk Licensing and Infrastructure**

* **Splunk Licensing:**
  + Ensure your Splunk license covers expected data volume.
  + Calculate anticipated daily ingestion from AWS services.
    - Example: Each EC2 instance might generate 100MB/day of logs.
  + Choose Splunk Cloud or Splunk Enterprise based on infrastructure needs.
* **Infrastructure Requirements:**
  + **Splunk Enterprise:** Requires deployment of indexers, search heads, and heavy forwarders.
  + **Splunk Cloud:** Hosted by Splunk, reducing operational overhead.
* **Optional Features:**
  + Splunk Observability Cloud:
    - Provides advanced real-time observability with metrics, traces, and logs.
    - Integrated dashboards and AI-driven insights.

**Step 2.2: Splunk Add-ons for AWS**

Ensure you have the following Splunk components:

* **Splunk Add-on for AWS:** Facilitates data collection from AWS services.
* **Splunk App for AWS:** Provides pre-built dashboards and visualizations.

**Step 2.3: Validate Resource Availability**

* Verify network connectivity between AWS and Splunk (e.g., via VPN, VPC peering).
* Ensure sufficient compute and storage capacity for Splunk servers.

**3. Prepare AWS Environment**

**Step 3.1: Identify AWS Accounts and Regions**

* List all AWS accounts and regions where your infrastructure is deployed.
  + Example: Separate accounts for development, testing, and production.
  + Consider tools like AWS Organizations to manage multiple accounts.
* **Regional Strategy:**
  + If using multiple regions, determine whether to monitor globally or regionally.

**Step 3.2: Configure IAM Roles/Policies**

* **IAM Roles for Splunk Integration:**
  + Create a dedicated role in AWS with the required permissions for Splunk to access resources.
  + Use the principle of least privilege to restrict permissions.

**Phase 2: Setting Up Splunk Environment**

1. **Provision Splunk Infrastructure:**
   * Deploy Splunk Enterprise or Splunk Cloud for log ingestion and analytics.
   * Set up heavy forwarders if data processing is required.
2. **Install Required Add-ons:**
   * Install the **Splunk Add-on for AWS**:
     + Available on Splunkbase.
     + Supports CloudWatch, CloudTrail, ELB, S3, and VPC Flow Logs ingestion.
   * Install the **Splunk App for AWS**:
     + Pre-built dashboards for AWS services.
3. **Integrate AWS with Splunk:**
   * Set up AWS credentials in Splunk using IAM roles with required permissions.
   * Configure inputs for:
     + **CloudWatch Metrics** (performance data).
     + **CloudWatch Logs** (application and service logs).
     + **CloudTrail Logs** (audit logs).
     + **S3 Bucket** (for static file-based logs).
   * Use Kinesis Firehose for streaming logs into Splunk for real-time observability.

**Phase 3: Data Ingestion and Indexing**

1. **Configure Data Sources:**
   * Create data inputs for CloudWatch, CloudTrail, and other logs.
   * Ensure data parsing rules for accurate indexing (e.g., JSON parsing for structured logs).
2. **Set Up Indexing Strategy:**
   * Define separate indexes for different AWS services (e.g., aws\_logs, aws\_metrics).
   * Use appropriate retention policies to manage storage costs.
3. **Ensure Secure Connectivity:**
   * Use Splunk HEC (HTTP Event Collector) or other secure methods for data transfer.
   * Enable encryption for data at rest and in transit.

**Phase 4: Building Dashboards and Alerts**

1. **Custom Dashboards:**
   * Use the Splunk App for AWS as a baseline for pre-built dashboards.
   * Create custom dashboards for:
     + Infrastructure health.
     + Cost monitoring.
     + Service-specific metrics (e.g., Lambda errors, RDS performance).
2. **Define Alerts:**
   * Set up alerts for critical thresholds (e.g., high CPU usage, low disk space).
   * Use adaptive thresholds for services with dynamic workloads.
3. **Enable Traces and Logs Correlation:**
   * Correlate application traces with infrastructure logs.
   * Use distributed tracing tools like OpenTelemetry integrated with Splunk.

**Phase 5: Automation and Optimization**

1. **Automate Data Ingestion:**
   * Use AWS CloudFormation or Terraform for infrastructure as code (IaC).
   * Automate Splunk input configuration using Splunk REST API.
2. **Optimize Performance:**
   * Enable deduplication to reduce data ingestion costs.
   * Use Splunk's search and reporting optimizations.

**Phase 6: Monitoring and Scaling**

1. **Continuous Improvement:**
   * Regularly review metrics and KPIs.
   * Refine alerts to minimize noise and false positives.
2. **Scale as Needed:**
   * Expand to additional AWS accounts or regions.
   * Scale Splunk infrastructure based on data volume.
3. **Integrate with Incident Management:**
   * Connect Splunk alerts with tools like PagerDuty, Slack, or Jira.
   * Implement runbooks for incident resolution.

**NEWRLEIC**

New Relic offers a wide range of features that complement and enhance Splunk’s capabilities. While Splunk focuses heavily on log management and analytics, New Relic provides a more holistic observability platform with real-time monitoring for infrastructure, applications, and end-user experience. Here's a breakdown of additional features New Relic provides and how they complement or extend Splunk’s functionality:

**1. Infrastructure Monitoring**

* **Features:**
  + Deep insights into AWS resources such as EC2, RDS, S3, Lambda, ECS, and Kubernetes (EKS).
  + Integration with AWS CloudWatch for detailed metric collection.
  + Resource-level dashboards showing CPU, memory, disk, and network usage.
* **How It Enhances Splunk:**
  + Splunk primarily analyzes logs; New Relic adds real-time visualization of infrastructure performance metrics.
  + Proactive issue identification with automated thresholds and AI-driven anomaly detection.

**2. Application Performance Monitoring (APM)**

* **Features:**
  + Monitors the performance of applications in real-time.
  + Tracks requests, errors, and database calls.
  + Provides detailed transaction traces to identify bottlenecks.
  + Supported languages: Java, .NET, Python, Node.js, PHP, Ruby, and Go.
* **How It Enhances Splunk:**
  + Splunk can aggregate and analyze application logs, but New Relic APM provides deeper visibility into application performance, such as response times and code-level insights.

**3. Synthetic Monitoring**

* **Features:**
  + Simulates user traffic from various locations around the globe.
  + Tracks uptime, response times, and SLA compliance.
  + Detects performance issues before they affect real users.
* **How It Enhances Splunk:**
  + Adds proactive testing capabilities to monitor application availability and performance globally, which Splunk lacks.

**4. Distributed Tracing**

* **Features:**
  + Tracks requests across distributed systems and microservices.
  + Pinpoints performance issues in complex architectures.
  + Supports OpenTelemetry for custom trace instrumentation.
* **How It Enhances Splunk:**
  + While Splunk offers log correlation for tracing, New Relic provides a more intuitive interface and faster root-cause identification for traces across services.

**5. Logs in Context**

* **Features:**
  + Correlates logs with APM and Infrastructure metrics.
  + Allows developers to drill down from a log entry to the related transaction or infrastructure metric.
  + Supports ingestion from AWS CloudTrail, application logs, and custom sources.
* **How It Enhances Splunk:**
  + Integrates logs with metrics and traces, providing a unified view for faster troubleshooting.
  + Complements Splunk’s logging capabilities with contextual insights.

**6. Browser Monitoring**

* **Features:**
  + Tracks end-user performance metrics such as page load time, DOM processing, and time to first byte (TTFB).
  + Monitors user experience across different browsers and devices.
* **How It Enhances Splunk:**
  + Splunk does not provide direct browser-side performance monitoring, making New Relic essential for end-user experience tracking.

**7. Mobile Monitoring**

* **Features:**
  + Monitors performance and stability of mobile applications.
  + Tracks metrics like crashes, network performance, and user sessions.
  + Provides detailed crash diagnostics.
* **How It Enhances Splunk:**
  + Adds mobile application observability, a feature Splunk lacks.

**8. Kubernetes and Container Monitoring**

* **Features:**
  + Monitors containerized applications running on ECS, EKS, or Kubernetes clusters.
  + Provides visibility into pod health, resource utilization, and cluster-wide metrics.
  + Supports autoscaling insights and alerts.
* **How It Enhances Splunk:**
  + Splunk can analyze logs from Kubernetes, but New Relic provides detailed visualizations of pod and container performance.

**9. Serverless Monitoring**

* **Features:**
  + Specialized monitoring for AWS Lambda.
  + Tracks invocation count, duration, error rates, and cold start times.
  + Provides Lambda cost analysis.
* **How It Enhances Splunk:**
  + Offers deeper insights into serverless environments, which require more than log-based analysis.

**10. Real User Monitoring (RUM)**

* **Features:**
  + Tracks user interactions with web applications in real-time.
  + Analyzes session traces and user flows to identify performance issues.
  + Provides geographic and device-based performance data.
* **How It Enhances Splunk:**
  + Adds detailed visibility into user behavior and experience that Splunk does not natively offer.

**11. Alerting and Incident Response**

* **Features:**
  + AI-driven anomaly detection and incident management.
  + Customizable alerts based on thresholds or anomalies.
  + Integrated with tools like PagerDuty, Slack, and Jira.
* **How It Enhances Splunk:**
  + Splunk has robust alerting, but New Relic’s AI-powered features simplify anomaly detection and reduce noise.

**12. Dashboards and Custom Visualizations**

* **Features:**
  + Unified dashboards for metrics, logs, and traces.
  + Customizable views for AWS resources, applications, and users.
  + Shareable links for real-time collaboration.
* **How It Enhances Splunk:**
  + Splunk’s dashboards focus on log analysis, whereas New Relic’s dashboards provide multi-dimensional insights across metrics, logs, and traces.

**13. Workload and Entity Monitoring**

* **Features:**
  + Groups related infrastructure and services into workloads.
  + Tracks the health of workloads rather than individual entities.
  + Provides high-level views for teams or applications.
* **How It Enhances Splunk:**
  + Simplifies monitoring for teams by focusing on logical workloads instead of isolated logs.

**14. New Relic Lookout**

* **Features:**
  + AI-driven anomaly detection for infrastructure and applications.
  + Automatically identifies patterns and trends.
  + Provides actionable insights without manual configuration.
* **How It Enhances Splunk:**
  + Adds predictive insights and anomaly detection, which require more manual setup in Splunk.

**15. Security Monitoring**

* **Features:**
  + Monitors security events from CloudTrail, AWS Config, and other sources.
  + Identifies vulnerabilities and misconfigurations.
  + Offers compliance reporting.
* **How It Enhances Splunk:**
  + Complements Splunk’s security information and event management (SIEM) capabilities with focused AWS security insights.