**Observability Demo**

**Transcript**

<https://otter.ai/u/db1ZS3lk1tNDbK9saTnY8uVIVsg?view=transcript>

David Hope provided an overview of the Elastic Observability platform, highlighting new onboarding workflows for data integration, particularly for Kubernetes using OpenTelemetry. He demonstrated the ease of bringing logs, metrics, and traces into Elastic Observability through simple steps, including adding Helm repos and annotations. He also showcased cloud provider Quick Starts, such as AWS Firehose, and dashboards for AWS usage and Kubernetes metrics. The platform's AI assistant was featured for analyzing logs, metrics, and traces, and for anomaly detection. Additionally, he discussed the use of SLOs for health monitoring and LLM observability, integrating with Amazon Bedrock for detailed LLM performance metrics.

**Action Items**

* [ ] Investigate the "cannot access cart storage" error and work with Luca to fix the issue.
* [ ] Review the updated runbook in the GitHub repository for the cart service issue.

**Outline**

**Elastic Observability Platform Overview**

* David Hope introduces the Elastic Observability platform, highlighting its recent updates and features.
* The overview page in Elastic Observability displays alerts, log events, and hosts, raising the question of data onboarding.
* New onboarding workflows have been introduced to help users bring their data into Elastic Observability.
* Elastic has moved to using OpenTelemetry as the standard data collection methodology for Kubernetes data.

**Onboarding Kubernetes Data with OpenTelemetry**

* David explains the process of bringing Kubernetes data into Elastic Observability using OpenTelemetry.
* The steps involve adding the Helm repo, installing the OpenTelemetry operator, and adding annotations to Kubernetes namespaces or deployments.
* Cloud providers like AWS have new Quick Starts for data onboarding, such as using AWS Firehose to create a stream for logs and metrics.
* Elastic Observability provides out-of-the-box dashboards for Kubernetes data, displaying metrics like Node statuses, CPU usage, and memory usage.

**Elastic AI Assistant and Anomaly Detection**

* David introduces the Elastic AI assistant, which can analyze logs, metrics, and traces quickly.
* The AI assistant can provide insights into Kubernetes metrics, such as average memory usage over the last 15 minutes.
* The business health dashboard brings together logs, metrics, and traces data into a single view, showing revenue over time and error codes.
* Anomaly detection in the business health dashboard highlights issues like high average memory usage and service level objectives being impacted.

**Machine Learning Anomaly Detection**

* Elastic Observability uses machine learning for anomaly detection, providing early warning indicators.
* David discusses the benefits of machine learning over traditional observability tools, which can handle high cardinality data.
* The machine learning algorithms can detect anomalies in various data points, such as status codes, IP addresses, and visitor amounts.
* The AI assistant can analyze errors in real-time, providing recommendations and connecting to GitHub for additional context.

**Service Level Objectives (SLOs) and Error Budgets**

* David explains the concept of SLOs and error budgets, using the cart service SLO as an example.
* SLOs define the desired availability of a service, with error budgets indicating how much downtime is tolerable.
* When the error budget is exhausted, alerts can be set to notify of potential issues.
* The cart service SLO is currently performing poorly, with a target of 99.9% and an actual performance of 91%.

**Analyzing Service Performance and Errors**

* David examines the performance of the cart service over the last hour, noting variable latency and errors.
* The machine learning algorithm identifies anomalies compared to the previous week, triggering alerts.
* An error related to accessing car storage is identified, and the AI assistant provides recommendations and context from GitHub.
* The AI assistant can search GitHub for relevant data and bring it into the conversation, enhancing problem-solving capabilities.

**LLM Observability Integration**

* David introduces LLM observability, highlighting the integration with Amazon Bedrock.
* The integration provides insights into models used, invocations, tokens, errors, and token usage.
* Users can diagnose and resolve LLM issues by analyzing prompts, responses, and logs.
* Elastic Observability also offers tracing capabilities, helping to identify performance issues in LLM applications.

**Conclusion and Future Trends**

* David summarizes the key features of Elastic Observability, including data onboarding, analysis, machine learning, and AI assistant integration.
* SLOs are presented as a popular method for health monitoring.
* LLM observability is highlighted as a growing trend, with Elastic providing the necessary tools for integration and management.
* David concludes by emphasizing the importance of investing in LLM observability for future applications.