Application Support SOP for Apigee

Overview

Application support in Apigee involves various tasks such as proxy debugging, API/proxy restart, ELK log monitoring, and gathering performance metrics. As an Application Support Engineer, you should follow a Standard Operating Procedure (SOP) to effectively handle these tasks. Below is a general outline for the Application Support SOP in Apigee.

Proxy Debugging:

Identify the Issue:

Gather relevant information about the reported issue, including the API/proxy name, environment, and any error messages.

Reproduce the issue, if possible, to understand the problem better.

Analyze Logs:

Use the Apigee Edge UI or API to access the proxy logs.

Review the logs to identify any error messages, stack traces, or abnormal behavior.

Correlate the logs with the reported issue to narrow down the root cause.

Debug the Proxy:

Use the Apigee Edge Trace tool to capture and analyze the request/response flow of the proxy.

Step through the proxy flow to identify the specific point of failure or unexpected behavior.

Inspect the request/response headers, payloads, and policies to pinpoint the issue.

Implement and Test the Fix:

Based on the debugging findings, implement the necessary changes to the proxy configuration or policies.

Deploy the updated proxy to a test environment and verify that the issue is resolved.

Perform thorough testing to ensure the fix does not introduce any new issues.

API/Proxy Restart:

Identify the API/Proxy:

Determine the specific API or proxy that needs to be restarted.

Verify the environment (e.g., production, staging) where the restart should be performed.

Communicate with Stakeholders:

Notify the relevant stakeholders, such as the API owners or consumers, about the planned restart.

Provide an estimated downtime duration and any potential impact on dependent services.

Perform the Restart:

Use the Apigee Edge UI or API to initiate the restart of the API or proxy.

Monitor the restart process to ensure it completes successfully.

Verify that the API or proxy is accessible and functioning correctly after the restart.

Post-Restart Verification:

Perform basic testing to confirm that the API or proxy is responding as expected.

Monitor the logs and metrics for any anomalies or errors after the restart.

Communicate the successful restart to the stakeholders.

ELK Log Monitoring:

Configure Log Collection:

Ensure that the Apigee logs are being properly collected and ingested into the ELK stack.

Verify that the necessary log fields and formats are being captured.

Set Up Dashboards:

Create or update Kibana dashboards to visualize the Apigee logs effectively.

Include relevant filters, queries, and visualizations to monitor key metrics and events.

Monitor Logs:

Regularly monitor the Kibana dashboards for any anomalies, errors, or unusual patterns.

Set up alerts and notifications for critical events or thresholds.

Investigate and troubleshoot any issues identified through log monitoring.

Optimize Log Collection:

Continuously review and optimize the log collection process based on the evolving needs of the system.

Fine-tune log filters, parsing rules, and index mappings to improve log analysis and performance.

Gathering Performance Metrics:

Identify Key Metrics:

Determine the critical performance metrics to monitor, such as response times, throughput, error rates, and resource utilization.

Align the metrics with the business requirements and SLAs.

Configure Metric Collection:

Set up the necessary tools and integrations to collect performance metrics from Apigee.

Use Apigee Edge Analytics, custom policies, or third-party monitoring solutions to gather the required metrics.

Create Dashboards:

Develop dashboards using tools like Grafana or the Apigee Edge UI to visualize the performance metrics.

Create meaningful visualizations, such as charts, graphs, and tables, to represent the metrics effectively.

Analyze and Optimize:

Regularly analyze the performance metrics to identify trends, bottlenecks, or areas for improvement.

Collaborate with development teams to optimize the API/proxy performance based on the insights gained from the metrics.

Continuously monitor and refine the performance metrics to ensure they remain relevant and actionable.

Incident Management:

Incident Identification and Triage:

Monitor the logs, dashboards, and alerts to identify any incidents or issues.

Assess the severity and impact of the incident based on predefined criteria.

Prioritize the incident based on its urgency and potential business impact.

Incident Investigation and Resolution:

Follow a structured troubleshooting approach to investigate the incident.

Use the debugging techniques, log analysis, and performance metrics to identify the root cause.

Implement the necessary fixes or workarounds to resolve the incident.

Post-Incident Review:

Conduct a post-incident review to analyze the incident's impact, response effectiveness, and lessons learned.

Document the findings and recommendations for future improvements.

Update the relevant documentation, runbooks, or SOPs based on the post-incident review.

Continuous Improvement:

Regularly review and update the Application Support SOP based on new learnings, best practices, and evolving system requirements.

Foster a culture of knowledge sharing and collaboration within the team to improve the overall support process.

Participate in training and skill development activities to stay up-to-date with Apigee features and troubleshooting techniques.