Detecting Slow APIs (Latency > 2000ms)

How to Detect Slow APIs:  
On your Grafana dashboard, there’s a panel that tracks how long your API calls take. This panel shows the average response time as well as the 90th and 99th percentile response times. If any of these numbers go above 2000 milliseconds, that signals that some API calls are taking too long. The dashboard can be set to change colors—such as turning red—when the response times exceed this threshold, making it easy to spot slow performance.

Which Chart or Metric Indicates the Issue:  
The most important indicators are the percentile metrics (especially P90 and P99). When these values exceed 2000ms, it means a significant portion of your API calls are slower than they should be.

Detecting High Error Rates (>30%)

On the dashboard that includes a panel that calculates the percentage of API calls that end in errors. This panel shows the error rate as a percentage. If this percentage climbs above 30%, it means nearly one out of every three API calls is failing, which is a serious concern.

Which Metrics and Visualizations Highlight the Issue:  
The key metric here is the error rate.

Defining 5 Alerts for Service Monitoring

Here are five alerts you might set up to help keep your services running smoothly:

1. High API Latency Alert:  
   This alert would trigger if a significant portion of your API calls—especially those measured at the 99th percentile—take longer than 2 seconds to respond.
2. High Error Rate Alert:  
   This alert activates if the percentage of failed API calls goes above 30% over a certain time period, indicating that too many requests are failing.
3. Low Request Throughput Alert:  
   This alert would go off if the number of API requests per second drops below an expected level, suggesting that your service might not be handling as many requests as it normally should.
4. Service Unavailability Alert:  
   This alert notifies you if one of your services stops sending data to your monitoring system, which could mean that the service is down or unreachable.
5. Resource Exhaustion Alert:  
   This alert is triggered if a service’s CPU or memory usage goes beyond a critical level, which could lead to performance issues or even crashes.

Runbook for the High Error Rate Alert:

1. Verify the Alert: When you get an alert, first check your dashboard to confirm that the error rate is indeed above 30%.
2. Check Logs and Metrics: Look at the application logs for any error messages or unusual activity, and review other metrics like response times and resource usage to see if there’s a broader problem.
3. Investigate: Make sure that all other services your application depends on—such as external APIs—are working correctly, as an issue with one of these can cause a spike in errors.
4. Take Mitigation Actions: If it looks like a temporary glitch, consider restarting the affected service. If the problem persists, you might need to scale up your resources or roll back any recent changes.
5. Document the Incident: After resolving the issue, record what happened, the steps you took, and any lessons learned. This documentation will help prevent similar problems in the future.