Designing an API Gateway Strategy

Reza Salehi
MCSE (CLOUD PLATFORM AND INFRASTRUCTURE), MCT, MCPD

@zaalion linkedin.com/in/rezasalehi2008

Overview

Understanding the purpose of an API gateway

Reviewing APIM instance limitations

Improving backend API performance

APIM logging and auditing

- Integration with Application Insights
- Integration with Azure Monitor

Demo: Azure Monitor and Application Insights integration with APIM

Summary

API Gateways and APIM

The Purpose of an API Gateway

Decouple Clients from Services

Use the gateway to route requests to one or more backend services

Gateway Offloading

Use the gateway to offload functionality from individual services to the gateway

Decouple Clients from Services

Use the gateway to aggregate multiple individual requests into a single request

Backend services don't need to expose a client-friendly protocol such as HTTP or WebSocket

Backend services are not publicly exposed so they are not potential attack surfaces

The client does not need to know how the individual backend services are structured

Gateway Offloading

Authentication, Web application firewall

IP white/blacklisting

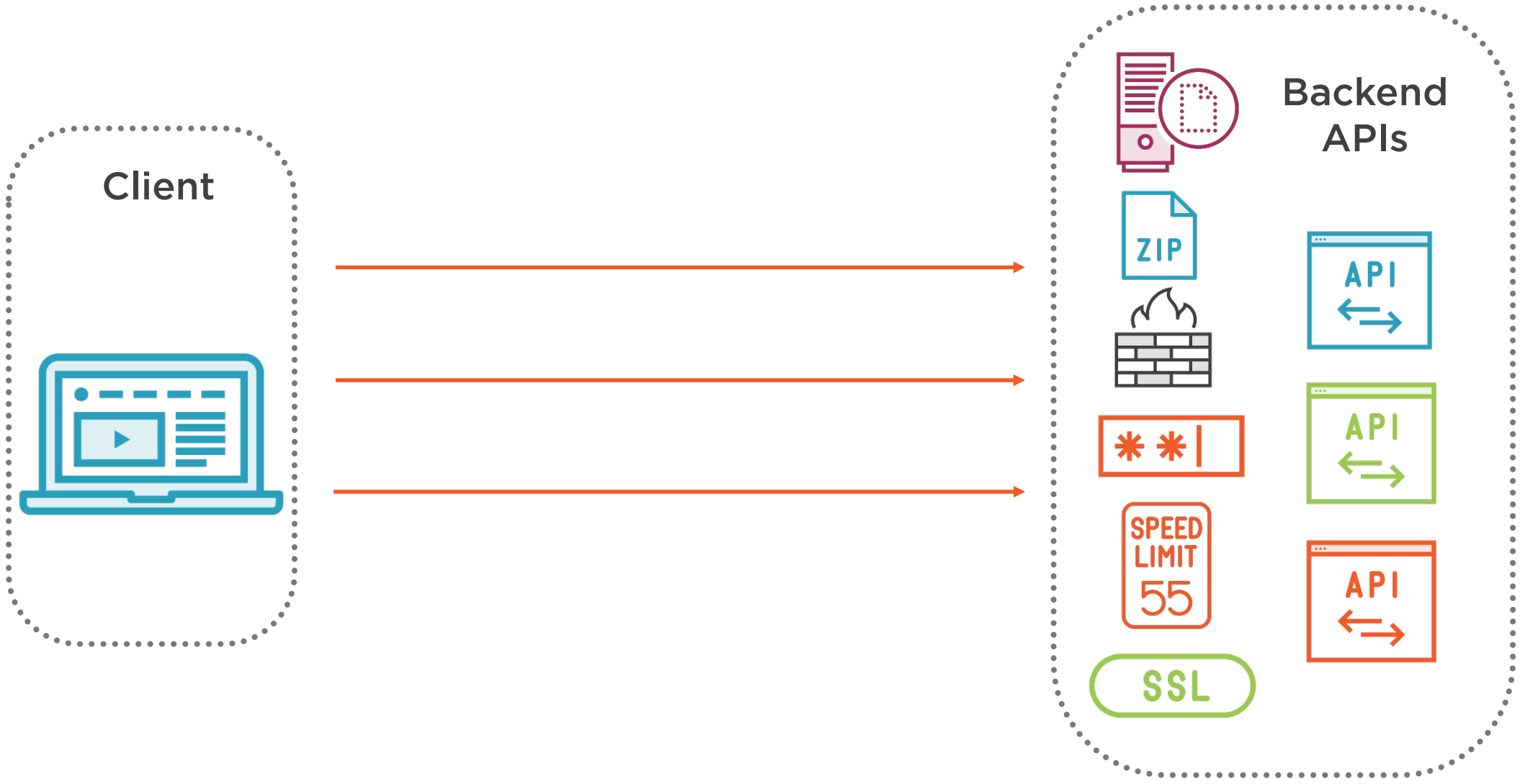
Client rate limiting (throttling)

Logging and monitoring

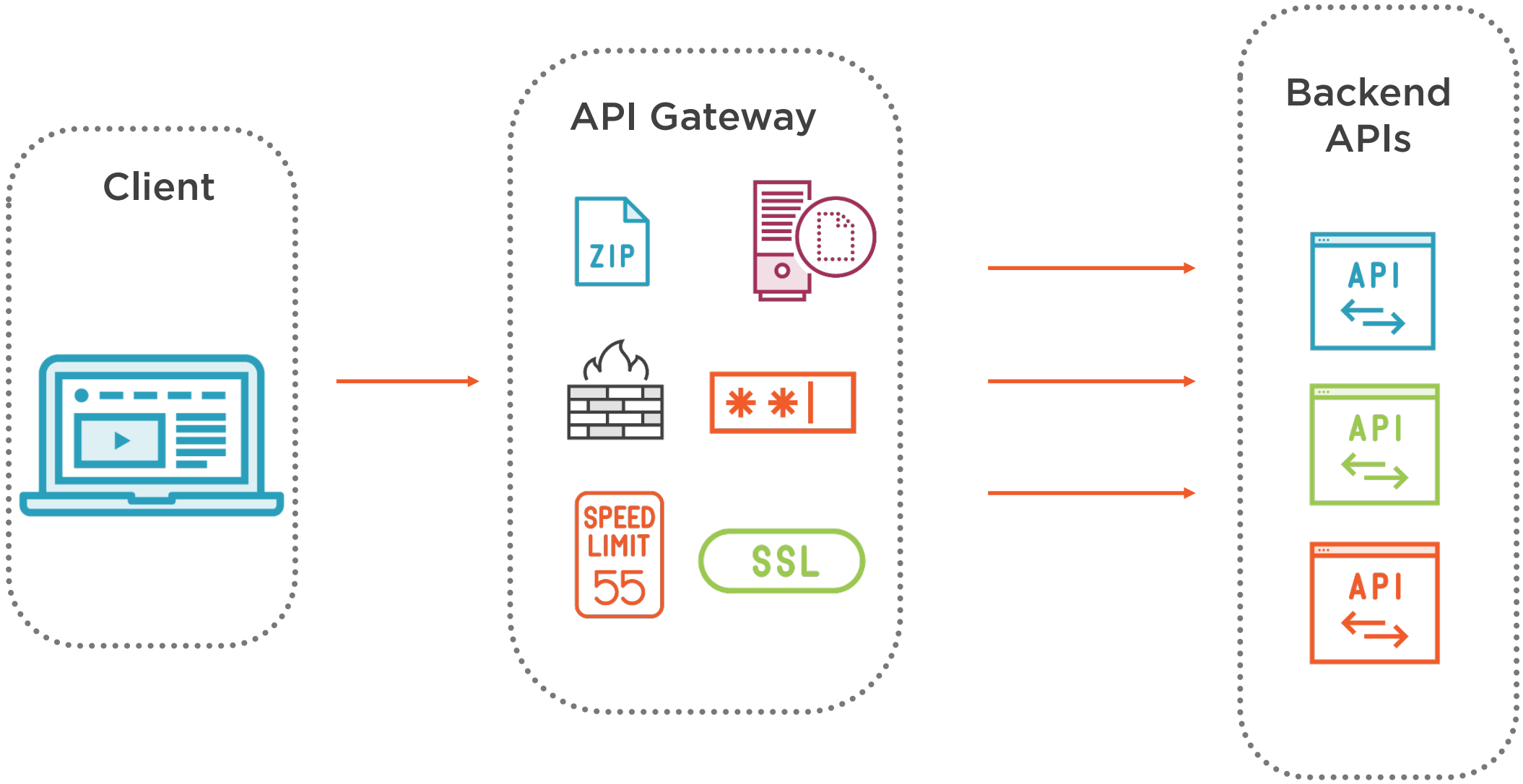
Response caching, Servicing static content

Compression

The Purpose of an API Gateway



The Purpose of an API Gateway



Improving Your API Performance

Common Antipatterns

Extraneous-fetching Antipattern

Retrieving more data than needed

Chatty I/O Antipattern

A large number of I/O requests

Extraneous-fetching Antipattern

Avoid fetching large volumes of data that may quickly become outdated or might be discarded

Implement pagination and only fetch a limited number of entities at a time

Take advantage of features built into the data store. SQL databases provide aggregate functions

Examine the source code to determine whether all of these fields are actually necessary

Chatty I/O Antipattern

Reduce the number of I/O requests by packaging the data into larger, fewer requests

When writing data, avoid locking resources for longer than necessary

Consider caching data that you retrieve from a service or a database

API Management Limits

API Management limits

Resource	Limit
Units of scale	10 per region[1]
Cache	5 GB per unit[1]
Concurrent backend connections[2] per HTTP authority	2048 per unit[3]
Maximum cached response size	10MB
Maximum policy document size	256KB
Maximum custom gateway domains	20 per service instance[4]

[1]API Management limits are different for each pricing tier. To see the pricing tiers and their scaling limits go to API Management Pricing. [2] Connections are pooled and re-used, unless explicitly closed by the backend. [3] Per unit of Basic, Standard and Premium tiers. Developer tier is limited to 1024. [4] Available in Premium tier only.

APIM Logging and Monitoring

Azure Application Insights

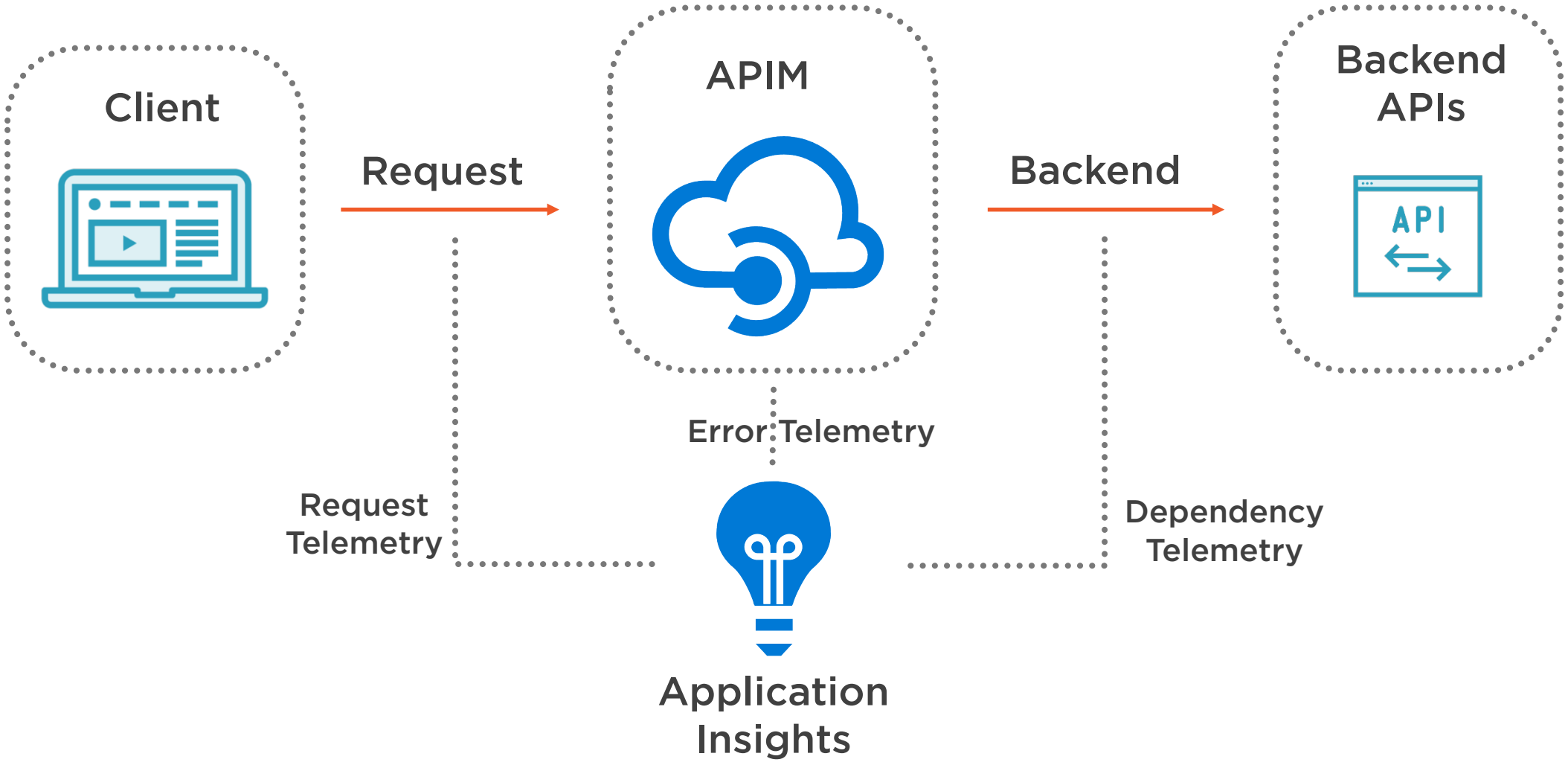
Azure Application Insights receives:

- Request telemetry, for every incoming request
- Dependency telemetry, for every request forwarded to a backend service
- Exception telemetry, for every failed request

A failed request is a request, which:

- Failed because of a closed client connection
- Triggered an on-error section of the API policies
- Has a response HTTP status code matching 400 or 500 family

APIM and Azure Application Insights



Performance Implications of Application Insights

Based on internal load tests, enabling this feature caused a 40%-50% reduction in throughput when request rate exceeded 1,000 requests per second

Manipulate the number of requests being logged by adjusting the Sampling setting. Value 100% means all requests are logged, while 0% reflects no logging at all. Sampling helps to reduce volume of telemetry

Skipping logging of headers and body of requests and responses will have positive impact on performance

Azure Application Insights is designed for assessing application performances.

It is not intended to be an audit system and is not suited for logging each individual request.

APIM and Azure Monitor

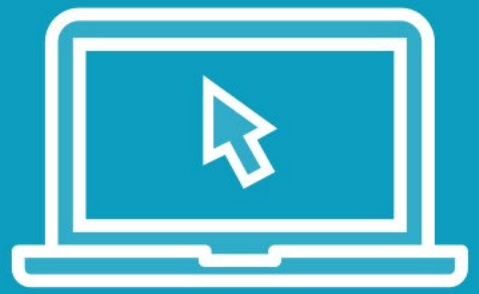
View metrics of your APIs, giving you near real-time visibility into the state and health of your APIs

You can configure alerts based on metrics and activity logs (email, webhook, LogicApp)

Activity logs provide insight into the operations that were performed on your API Management services

Diagnostic logs provide information about operations and errors that are important for auditing and troubleshooting

Demo



Integrating APIM with Azure Application Insights

Integrating APIM with Azure Monitor

Summary

- Understanding the purpose of an API gateway
  - Specifically APIM
- Listing APIM instance limitations
- A few notes to improve backend API performance
- Integration with Application Insights
- Integration with Azure Monitor
- Demo: Azure Monitor and Application Insights integration with APIM
- Summary