{adobe.io}

open source
https://github.com/adobe-apiplatform

dragos dascalita haut | project lead | adobe.io

Docker Container with Openresty + API Gateway

Source: https://github.com/adobe-apiplatform/apigateway

What's special:

- tiny - based on Alpine Linux **~ 100MB**
  - https://hub.docker.com/r/adobeapiplatform/apigateway/
- bundles the openresty/nginx **debugger version** too
- plan is to add more debugging tools like CPU flamegraphs

api-gateway-request-validation

Source: https://github.com/adobe-apiplatform/api-gateway-request-validation

Simple validation framework for requests based on sub-requests.

Declarative, built on top of the NGINX config :

```
set $validate_api_key “on; path=/validate_api_key; order=1; ”;

set $validate_service_plan “on; path=/validate_service_plan; order=1; “;

access_by_lua ‘ngx.apiGateway.validation.validateRequest()’;
```

api-gateway-async-logger

Source: COMING SOON

- currently supporting Kinesis
- Kafka support is underway

This module leverages the Openresty AWS SDK already in the open-source space: https://github.com/adobe-apiplatform/api-gateway-aws

Can easily send 10,000 request per second from a single node using 10%CPU, 63.5Mbps and 16MB shared dict form a docker container with 8 CPUs, 8GB Memory.

ZMQ Adapter

Source: https://github.com/adobe-apiplatform/api-gateway-zmq-adaptor

ZMQ Logger based on FFI: https://github.com/adobe-apiplatform/api-gateway-zmq-logger

```
local zmqLogger = ZmqLogger:new()
zmqLogger:connect(ZmqLogger.SOCKET_TYPE.ZMQ_PUB, zmq_publish_address)
zmqLogger.log("hello-world")
```

Can send hundreds of thousands of requests per second.

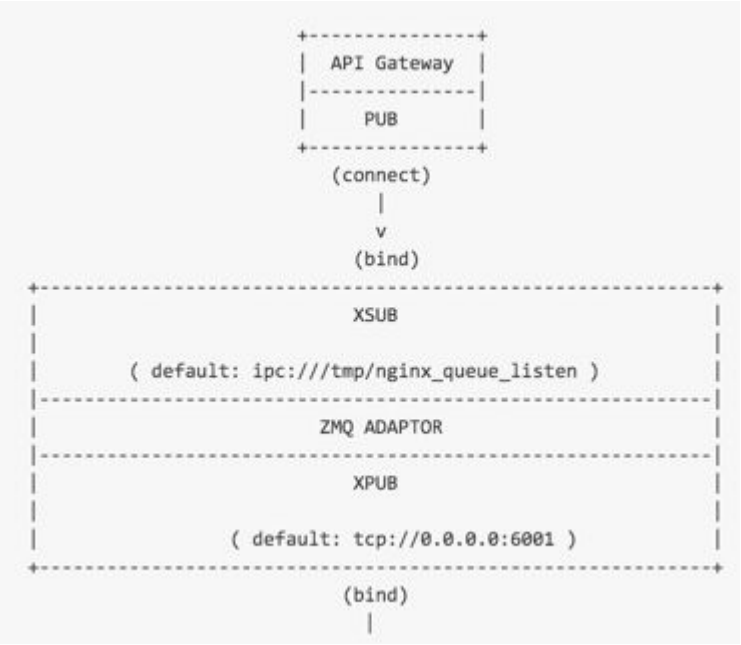
Throttling/rate limiting - ASYNC

Source: COMING SOON.

Based on the ZMQ adapter.

Key feature: ASYNC

- usage data is sent Async via ZMQ to a microservice
- it doesn’t block the request



api-gateway-cache-manager

Source: https://github.com/adobe-apiplatform/api-gateway-cachemanager

Library for managing multiple cache stores (local cache store, Redis cache store)

It can leverage existing modules (srcache, shared_dict, lru_cache)

Demo:

API Gateway -> validation -> AWS Lambda -> AWS Kinesis -> ElasticSearch