API Gateway Blueprint

Thor Chin
(thor.c@inwinstack.com)
March 21, 2019
## Blueprint Proposal: API Gateway

<table>
<thead>
<tr>
<th>Case Attributes</th>
<th>Description</th>
<th>Informational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>New Blueprint</td>
<td></td>
</tr>
<tr>
<td>Blueprint Family - Proposed Name</td>
<td>Not belong to blueprint family</td>
<td></td>
</tr>
<tr>
<td>Use Case</td>
<td>Provide API Gateway for OpenStack, K8S and other platform or applications</td>
<td></td>
</tr>
<tr>
<td>Blueprint proposed Name</td>
<td>API Gateway</td>
<td></td>
</tr>
<tr>
<td>Initial POD Cost (capex)</td>
<td>1 node or 2 nodes for HA</td>
<td></td>
</tr>
<tr>
<td>Scale &amp; Type</td>
<td>Up to 2 Arm/x86 Servers</td>
<td></td>
</tr>
<tr>
<td>Applications</td>
<td>API gateway for different platform integration. For example, api for create K8S containers and VMs.</td>
<td></td>
</tr>
<tr>
<td>Power Restrictions</td>
<td>depends on hardware</td>
<td></td>
</tr>
<tr>
<td>Infrastructure orchestration</td>
<td>OpenStack Queens or above - VM orchestration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Docker 1.13.1 or above and K8s 1.10.2 or above- Container Orchestration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OS - Ubuntu 16.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Under Cloud Orchestration - Airship v1.0</td>
<td></td>
</tr>
<tr>
<td>SDN</td>
<td>OVS, Calico</td>
<td></td>
</tr>
<tr>
<td>Workload Type</td>
<td>Containers, VMs</td>
<td></td>
</tr>
<tr>
<td>Additional Details</td>
<td>Kong</td>
<td></td>
</tr>
</tbody>
</table>
Motivation

- Complexity for client to connect to multiple microservices
- Mixing communication protocols such as REST, SOAP, XML-RPC
- Authentication and Authorization for multiple microservices
- Prevent DDoS or SQL-injection attack
- Monitoring for the API is used by whom and when or how
- Micro service IP may change and need to update all API connection info.
API Gateway Architecture

Web Client

Mobile Client

API Gateway

DNS service

Microservice 1

Microservice 2

Microservice 3

Back end
Benefits of API Gateway

- Security (Authentication and authorization)
- Load Balancing
- Lifecycle Management
- Ease to monitoring
- Service discovery (IP change Issue)

Before

Web Client
- Log service
- Monitoring service
- Updating service
- Customer service

Mobile Client

After

Web Client
- API Gateway
- DNS service
- getLog()
- getMonInfo()
- getProductInfo()
- getUpdateInfo()

Mobile Client
- Log service
- Monitoring service
- Updating service
- Customer service
Technical Architecture

Client / Server API

Authentication
ACL
Monitoring
Caching
Logging
Rate-Limiting
Security
Serverless

API Gateway

API / RPC
Private

API / RPC
Public

API / RPC
Partner

Can be set on Container / VM / x86 Server / ARM Server