Run Apache APISIX on Kubernetes

Jintao Zhang

Me?

Jintao Zhang

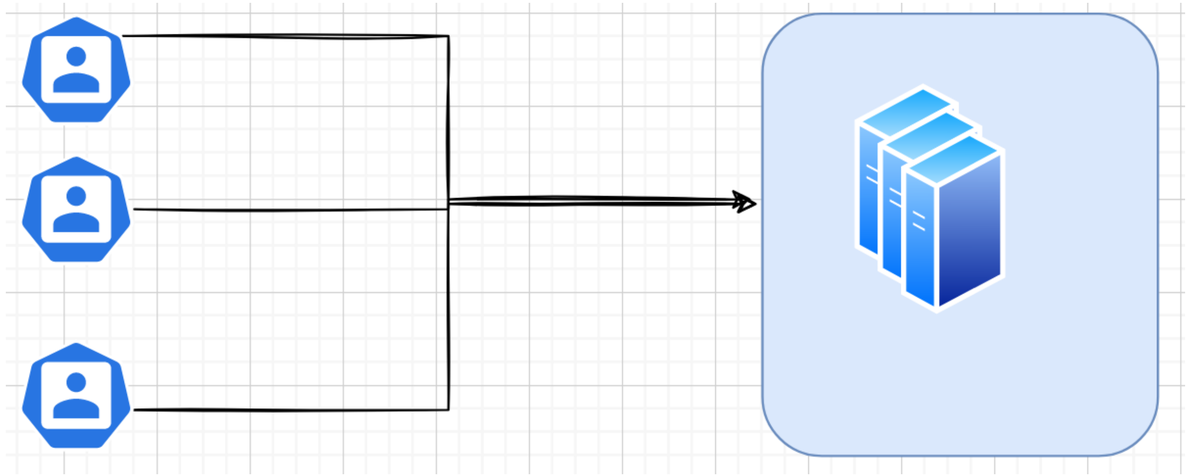
- Apache APISIX committer
- Kubernetes ingress-nginx reviewer
- GitHub: tao12345666333
- Mail: zhangjintao@apache.org

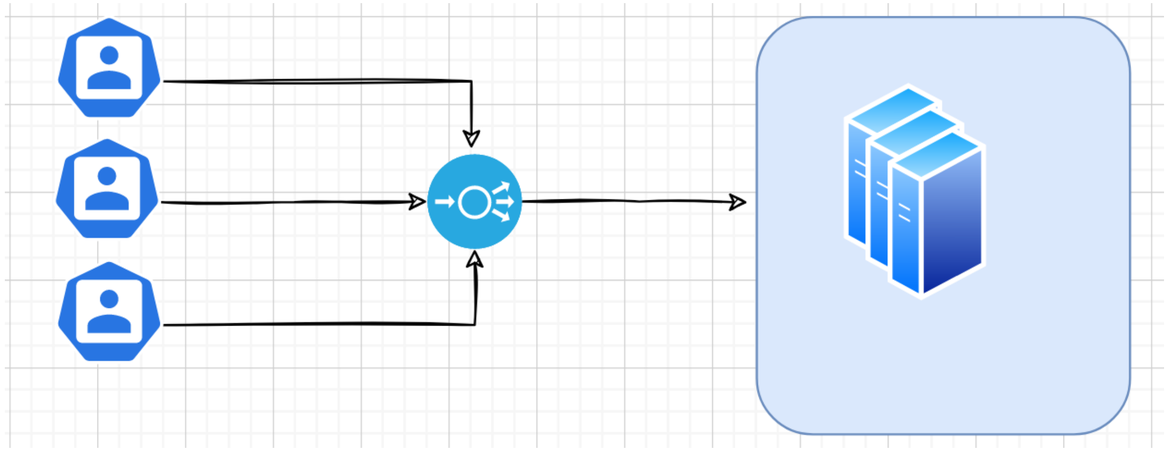
Agenda

- API Gateway 101
- What is Apache APISIX
- Running Apache APISIX as Gateway
- Running Apache APISIX as Ingress controller
- Summary

What is Proxy

- Forward proxy （mask client）
- Reverse proxy （mask server）
- Reverse proxy vs LB





What is API Gateway

- Unified traffic ingress and egress between client and backend
- API management center
- Traffic topology management

Why we need API Gateway

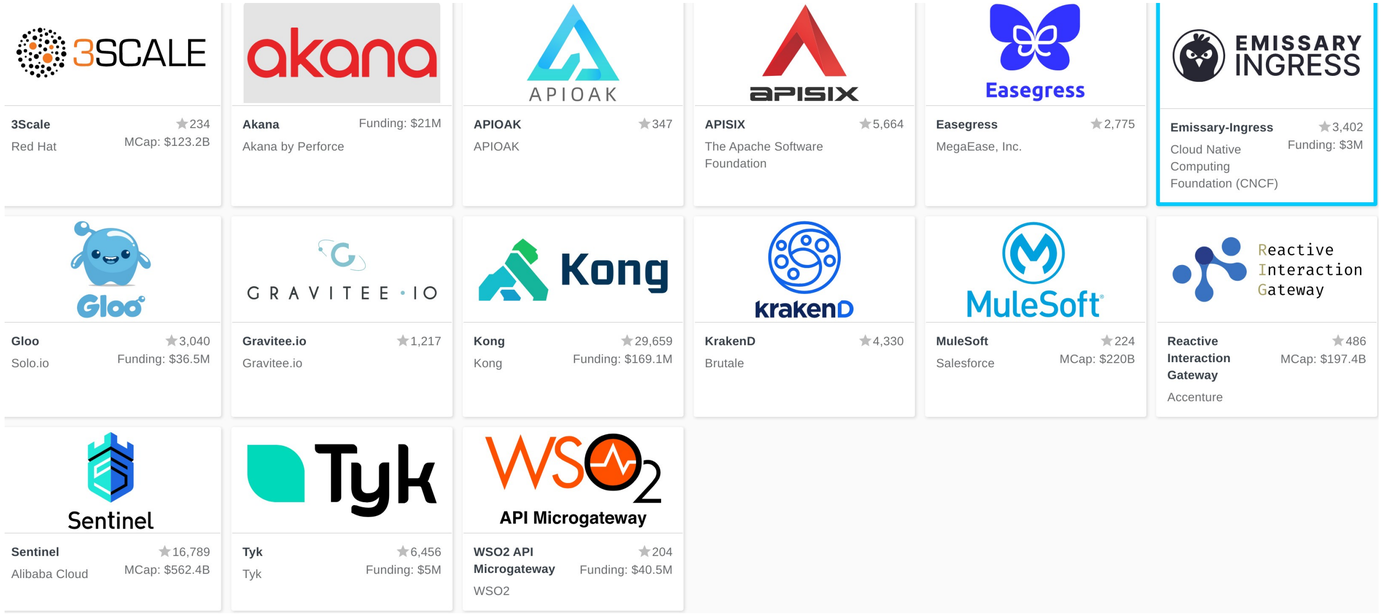
- Increased business complexity
- Popularity of microservice-based systems
- Abstraction layer for backend microservices
- Authentication / Authorization
- Rate limiting / Circuit Breaker

API Gateway core functions

- Dynamic API management
- Load Balancing
- Authentication / Authorization
- Traffic management
- Observability / Analyzing

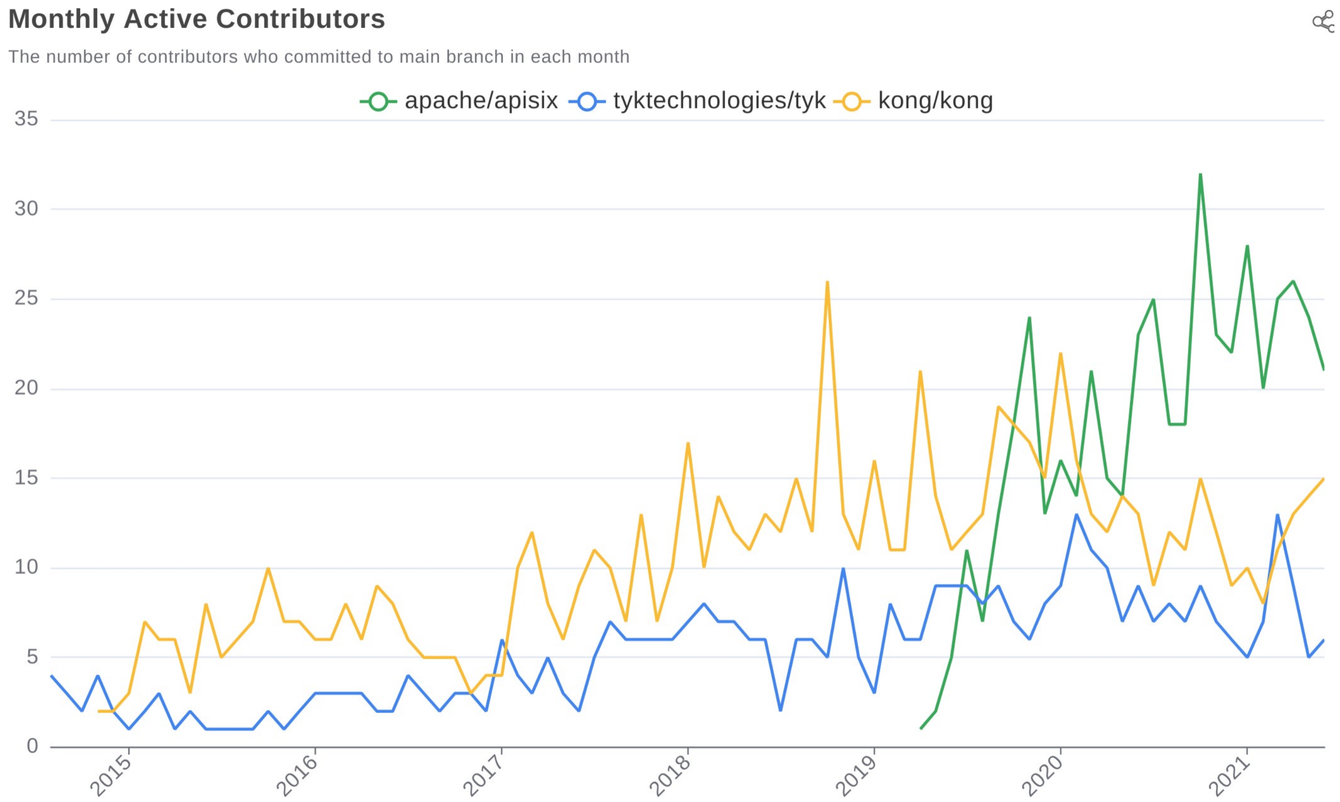
API Gateway landscape

- Different technology stacks
- Different levels of activity
- Different performance



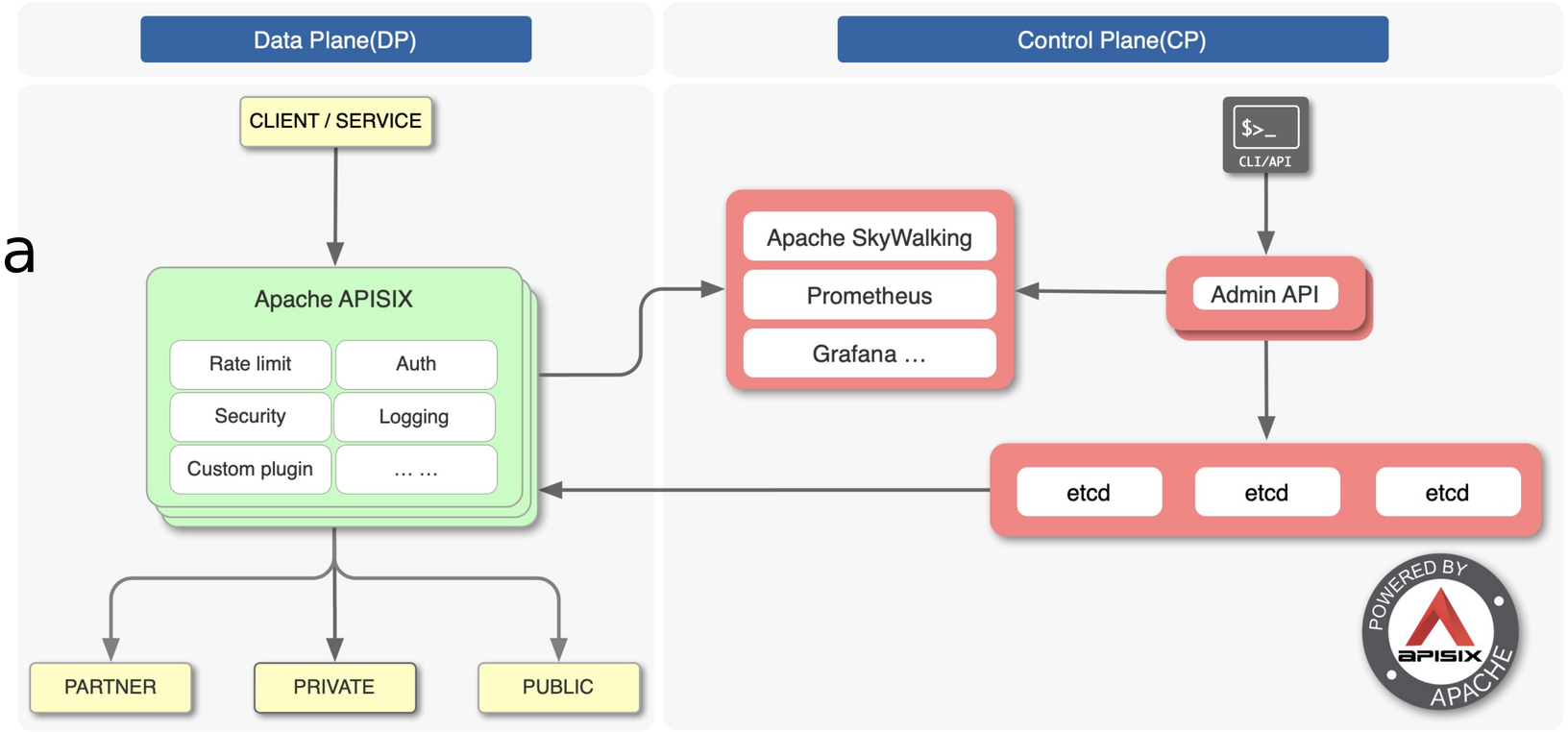
What is Apache APISIX

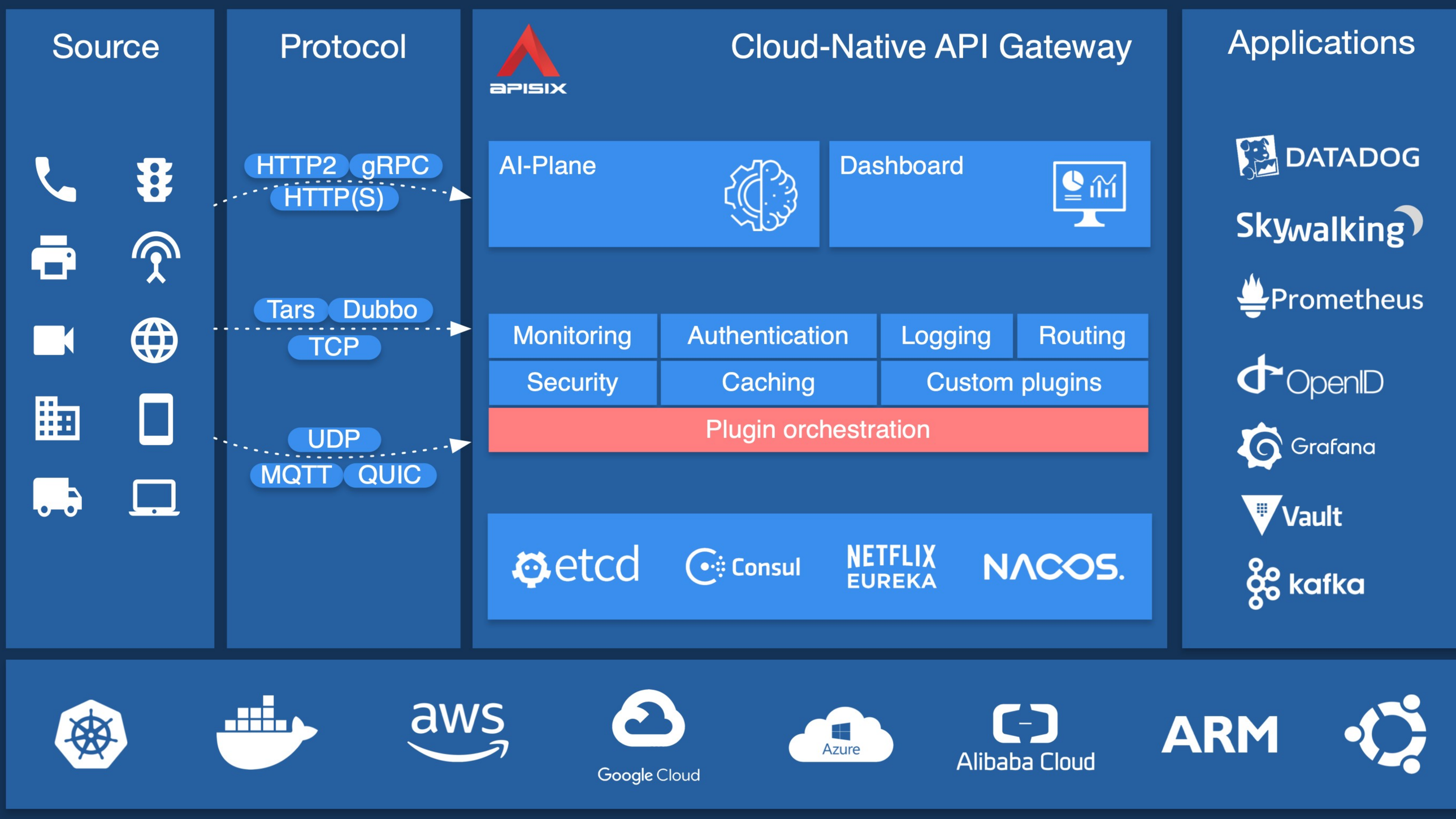
- ASF top-level project
- Cloud-native API Gateway
- High performance
- Fully dynamic
- Most active Open Source API Gateway Project



Architecture overview

- Base on NGINX + lua
- etcd as storage




Source
Protocol
APISIX
Cloud-Native API Gateway
Applications
HTTP2
gRPC
HTTP(S)
Tars
Dubbo
TCP
UDP
MQTT
QUIC
AI-Plane
Dashboard
Monitoring
Authentication
Logging
Routing
Security
Caching
Custom plugins
Plugin orchestration
etcd
Consul
NETFLIX EUREKA
NACOS.
DATADOG
Skywalking
Prometheus
OpenID
Grafana
Vault
kafka
aws
Google Cloud
Azure
Alibaba Cloud
ARM

How to run Apache APISIX

- Binary
- Docker image

  https://hub.docker.com/u/apache/apisix
- Kubernetes (Helm chart)

  https://github.com/apache/apisix-helm-chart

Deploy httpbin demo

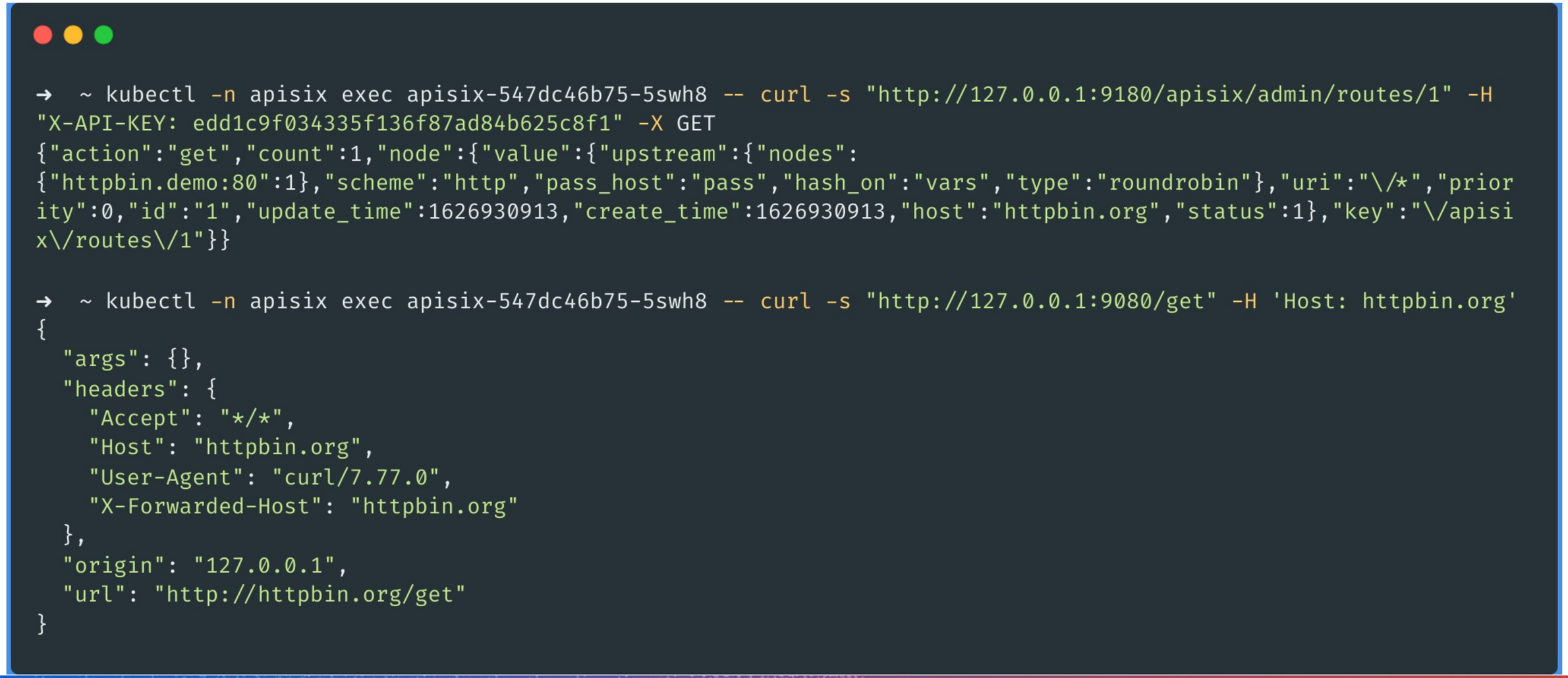
```
➜  ~ kubectl create ns demo
namespace/demo created
➜  ~ kubectl -n demo run httpbin --image kennethreitz/httpbin --port 80
pod/httpbin created
➜  ~ kubectl -n demo expose pod httpbin --port 80
service/httpbin exposed
➜  ~ kubectl -n demo get pods,svc
NAME          READY   STATUS    RESTARTS   AGE
pod/httpbin   1/1     Running   0          1m

NAME              TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
service/httpbin   ClusterIP   10.96.118.8    <none>        80/TCP    1m
```

Deploy Apache APISIX

```
➜  git clone https://github.com/apache/apisix-helm-chart.git
➜  cd apisix-helm-chart
➜  apisix-helm-chart git:(master) kubectl create ns  apisix
➜  apisix-helm-chart git:(master) helm install apisix charts/apisix --set gateway.type=NodePort --namespace
apisix
...
➜  apisix-helm-chart git:(master) kubectl -n apisix exec apisix-547dc46b75-5swh8 -- curl -s
"http://127.0.0.1:9180/apisix/admin/routes/1" -H "X-API-KEY: edd1c9f034335f136f87ad84b625c8f1" -X PUT -d '
{
  "uri": "/*",
  "host": "httpbin.org",
  "upstream": {
    "type": "roundrobin",
    "nodes": {
      "httpbin.demo:80": 1
    }
  }
}'
{"node":{"value":{"upstream":{"nodes":
{"httpbin.demo:80":1},"scheme":"http","type":"roundrobin","hash_on":"vars","pass_host":"pass"},"update_time":1626
930913,"priority":0,"id":"1","status":1,"create_time":1626930913,"host":"httpbin.org","uri":"\/*"},"key":"\/apisi
x\/routes\/1"},"action":"set"}
```

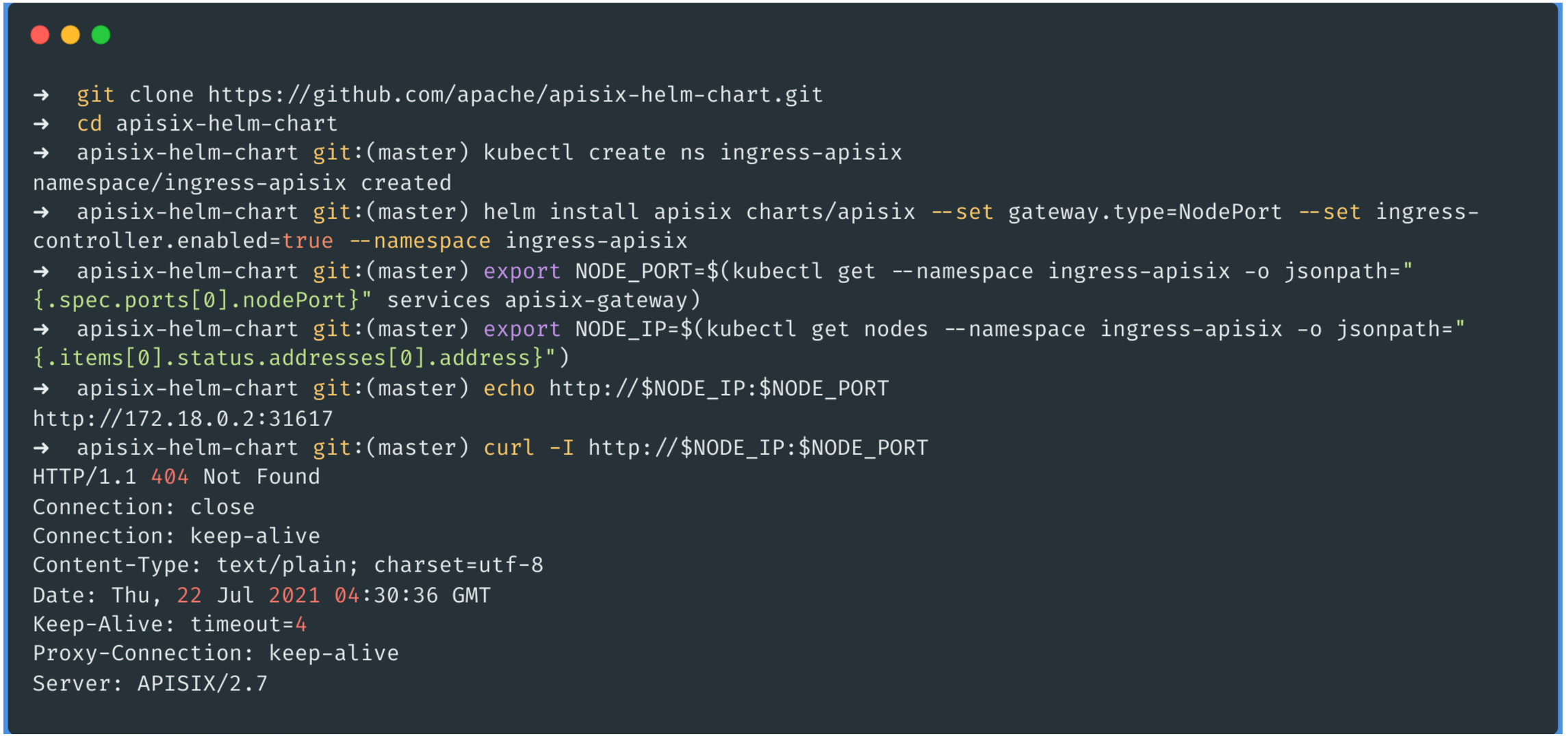
Visit demo using APISIX



```
➜  ~ kubectl -n apisix exec apisix-547dc46b75-5swh8 -- curl -s "http://127.0.0.1:9180/apisix/admin/routes/1" -H
"X-API-KEY: edd1c9f034335f136f87ad84b625c8f1" -X GET
{"action":"get","count":1,"node":{"value":{"upstream":{"nodes":
{"httpbin.demo:80":1},"scheme":"http","pass_host":"pass","hash_on":"vars","type":"roundrobin"},"uri":"\/*","prior
ity":0,"id":"1","update_time":1626930913,"create_time":1626930913,"host":"httpbin.org","status":1},"key":"\/apisi
x\/routes\/1"}}

➜  ~ kubectl -n apisix exec apisix-547dc46b75-5swh8 -- curl -s "http://127.0.0.1:9080/get" -H 'Host: httpbin.org'
{
  "args": {},
  "headers": {
    "Accept": "*/*",
    "Host": "httpbin.org",
    "User-Agent": "curl/7.77.0",
    "X-Forwarded-Host": "httpbin.org"
  },
  "origin": "127.0.0.1",
  "url": "http://httpbin.org/get"
}
```

Apache APISIX Ingress



```
➜  git clone https://github.com/apache/apisix-helm-chart.git
➜  cd apisix-helm-chart
➜  apisix-helm-chart git:(master) kubectl create ns ingress-apisix
namespace/ingress-apisix created
➜  apisix-helm-chart git:(master) helm install apisix charts/apisix --set gateway.type=NodePort --set ingress-
controller.enabled=true --namespace ingress-apisix
➜  apisix-helm-chart git:(master) export NODE_PORT=$(kubectl get --namespace ingress-apisix -o jsonpath="
{.spec.ports[0].nodePort}" services apisix-gateway)
➜  apisix-helm-chart git:(master) export NODE_IP=$(kubectl get nodes --namespace ingress-apisix -o jsonpath="
{.items[0].status.addresses[0].address}")
➜  apisix-helm-chart git:(master) echo http://$NODE_IP:$NODE_PORT
http://172.18.0.2:31617
➜  apisix-helm-chart git:(master) curl -I http://$NODE_IP:$NODE_PORT
HTTP/1.1 404 Not Found
Connection: close
Connection: keep-alive
Content-Type: text/plain; charset=utf-8
Date: Thu, 22 Jul 2021 04:30:36 GMT
Keep-Alive: timeout=4
Proxy-Connection: keep-alive
Server: APISIX/2.7
```

Creating ApisixRoute

```
➜  ~ kubectl -n ingress-apisix run httpbin --image kennethreitz/httpbin --port 80
➜  ~ kubectl -n ingress-apisix expose pod httpbin --port 80
➜  ~ cat ingress-apisix.yaml
apiVersion: apisix.apache.org/v2alpha1
kind: ApisixRoute
metadata:
  name: httpserver-route
spec:
  http:
    - name: httpbin
      match:
        hosts:
          - ingress.httpbin.org
        paths:
          - "/get"
      backend:
        serviceName: httpbin
        servicePort: 80

➜  ~ kubectl -n ingress-apisix apply -f ingress-apisix.yaml
apisixroute.apisix.apache.org/httpserver-route configured
```

Apache APISIX Ingress

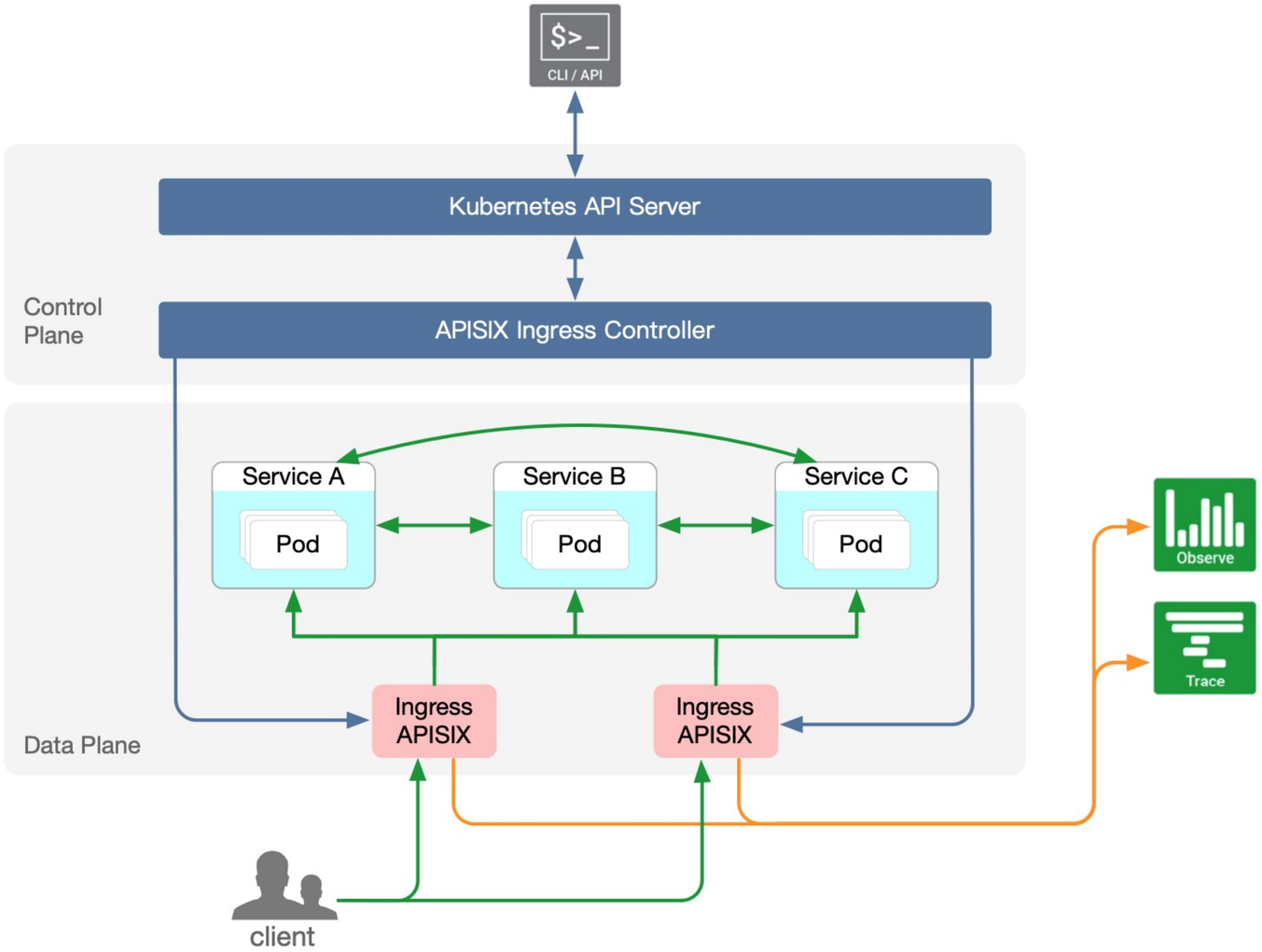
```
➜  ~ export NODE_PORT=$(kubectl get --namespace ingress-apisix -o jsonpath="{.spec.ports[0].nodePort}" services
apisix-gateway)
➜  ~ export NODE_IP=$(kubectl get nodes --namespace ingress-apisix -o jsonpath="
{.items[0].status.addresses[0].address}")
➜  ~ curl http://$NODE_IP:$NODE_PORT/get -H "HOST: ingress.httpbin.org"

{
  "args": {},
  "headers": {
    "Accept": "*/*",
    "Host": "ingress.httpbin.org",
    "User-Agent": "curl/7.76.1",
    "X-Forwarded-Host": "ingress.httpbin.org"
  },
  "origin": "10.244.0.1",
  "url": "http://ingress.httpbin.org/get"
}
```

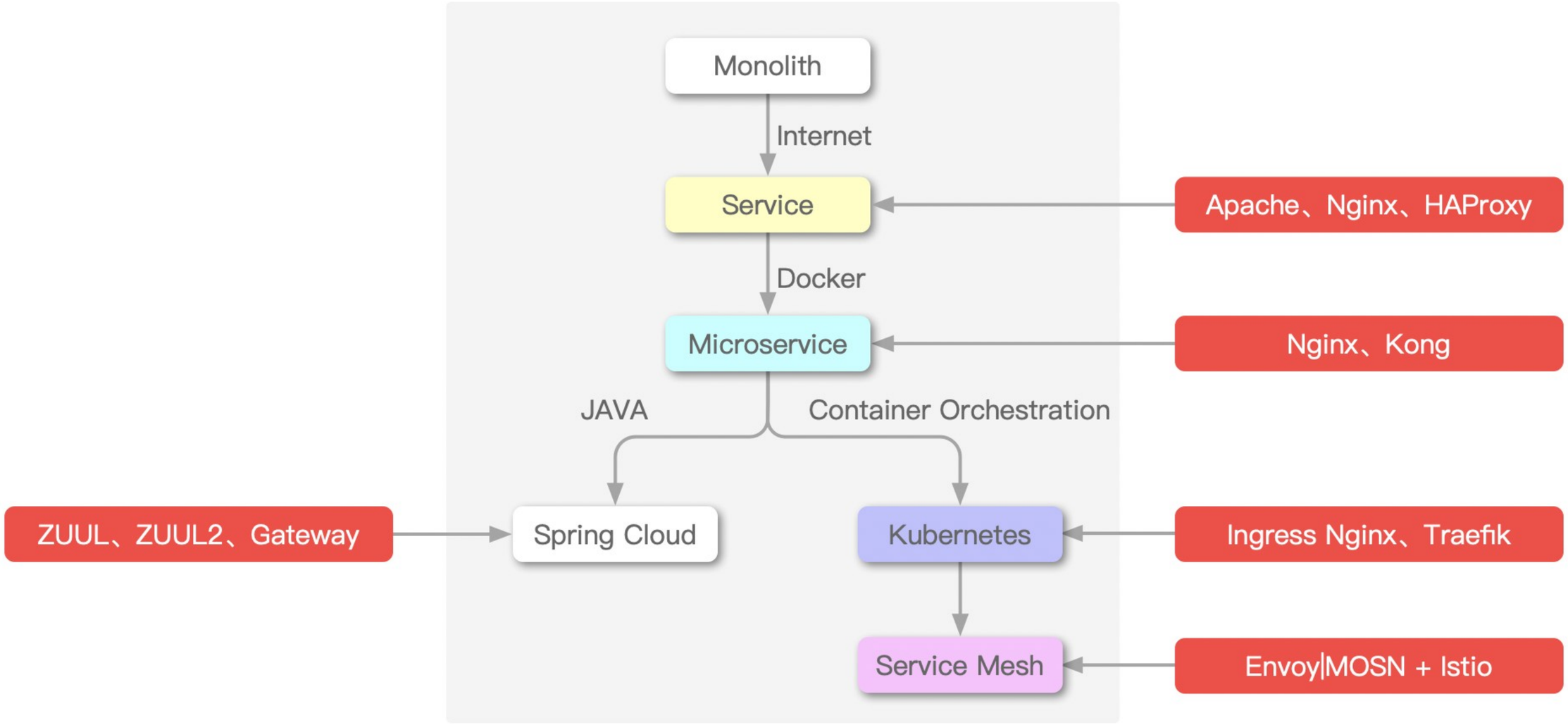
Apache APISIX Ingress

Apache APISIX Ingress

- High availability
- More flexible

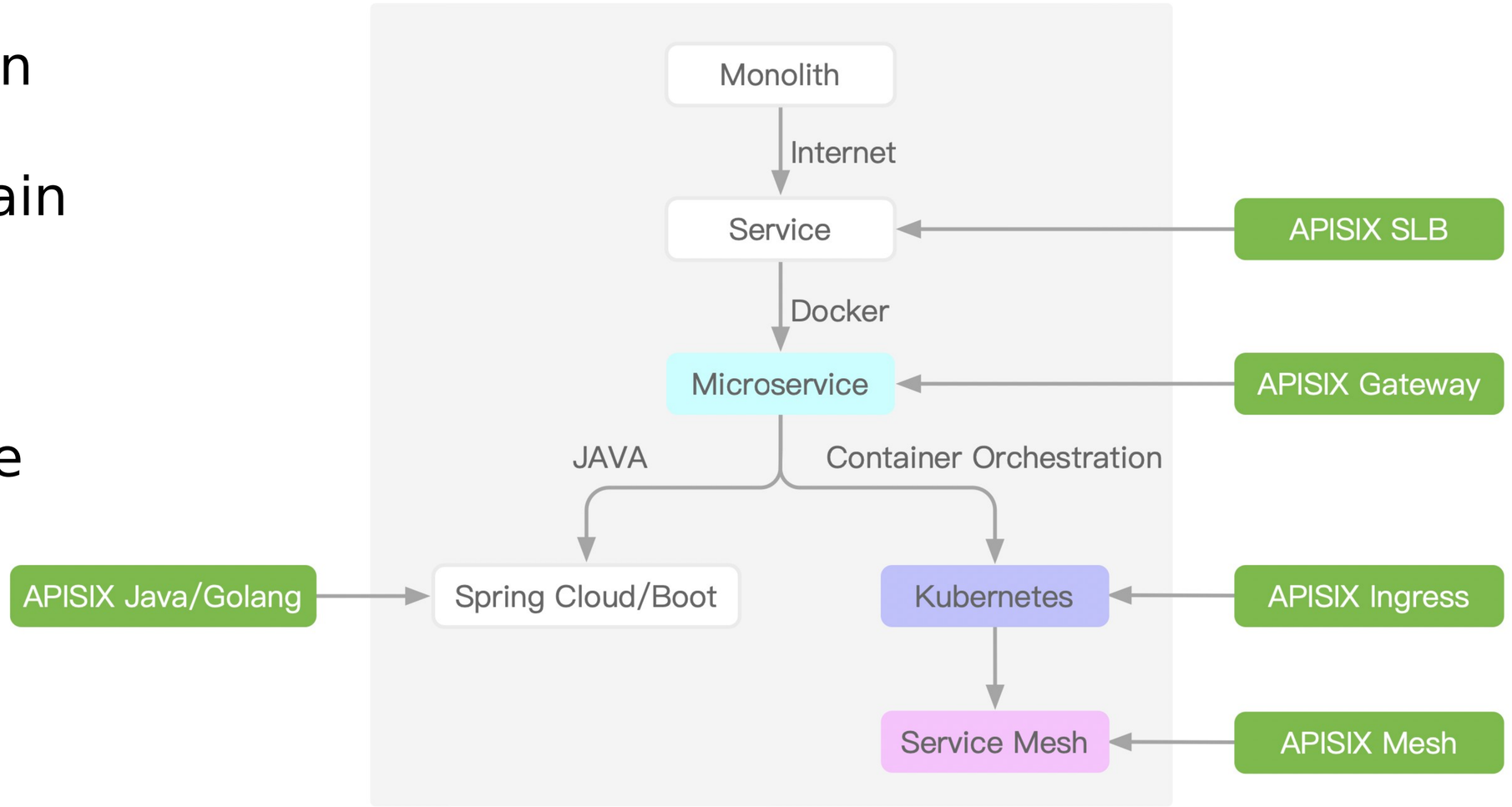


Complicated choice



Apache APISIX Way

- standardization
- Easy to maintain
- OPS friendly
- Highly scalable
- Full Dynamic



Summary

- What is API Gateway and why we need it
- What is Apache APISIX
- Run Apache APISIX on Kubernetes

`https://github.com/apache/apisix`

@zhangjintao9020

@ApacheAPISIX