API Programme Services (APS)

Client-Hosted Kong Gateway

Agenda

- What is a Client-Hosted Kong Gateway?
- Steps to Provisioning
- Demo

What is a Client-Hosted Kong Gateway

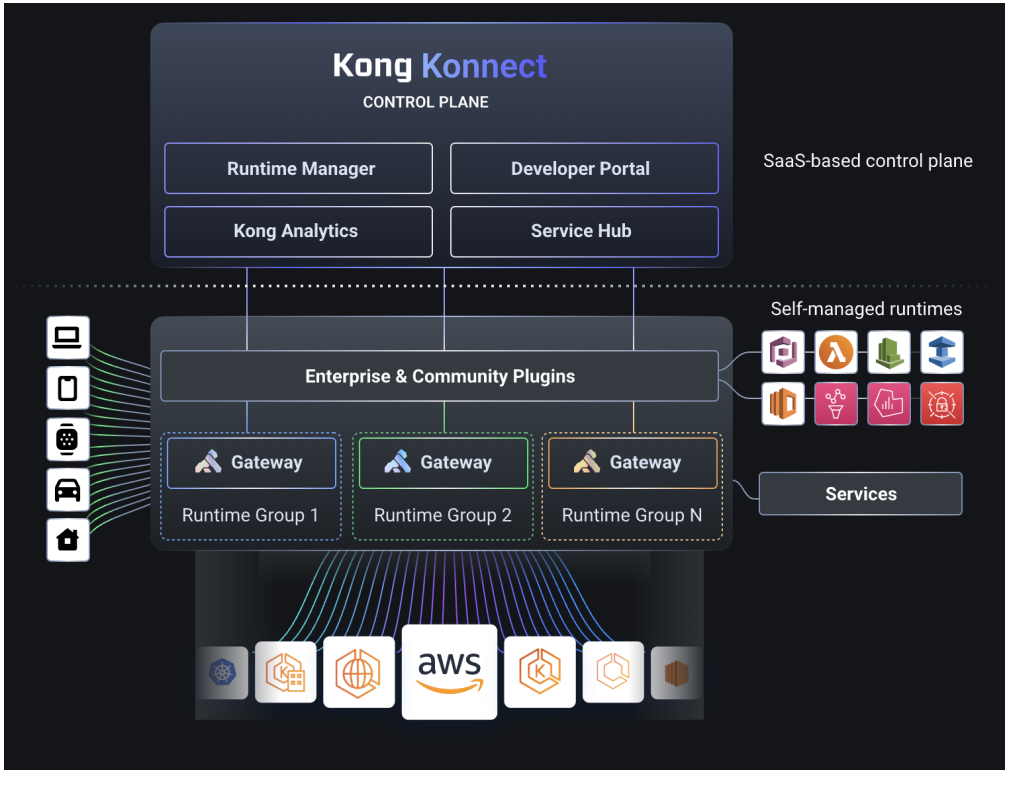
> The runtime engine, Kong Gateway - the fastest and most adopted API gateway - runs within your preferred network environment providing ultimate security and efficiency at scale. For ~~Organizations~~ Ministries, this means they can focus on shipping applications better, faster, and in a more secure way to deliver transformational digital experiences.

The control plane is hosted in Openshift by APS, while the runtimes, Kong Gateway, run within your preferred environment.

APS Services:

- Credential lifecycle management
- Developer Portal
- RBAC access control
- Audit
- Security Information Event Monitoring (SIEM) *
- Configuration Management
- Monitoring / Analytics about your APIs (Performance, Traffic)

Kong Konnect



Steps to Provisioning a Kong Gateway

1. Request a new Runtime Group
2. Acquire a Domain Name and SSL certificate
3. Deploy Kong Gateway
4. Update DNS
5. Configure your Services
6. Manage your APIs

Request a new Runtime Group

A group of instances of Kong Gateway is called a Runtime Group

APS has some initial setup to support a new Runtime Group

APS hosts two Runtime Groups: OCP Silver cluster and OCP Gold cluster

Acquire a Domain Name and SSL certificate

A new domain name and SSL cert must be obtained. Due to security reasons, APS's wildcard SSL certificates (`*.api.gov.bc.ca` and `*.apps.gov.bc.ca`) are not available for installation on a client-hosted Kong Gateway.

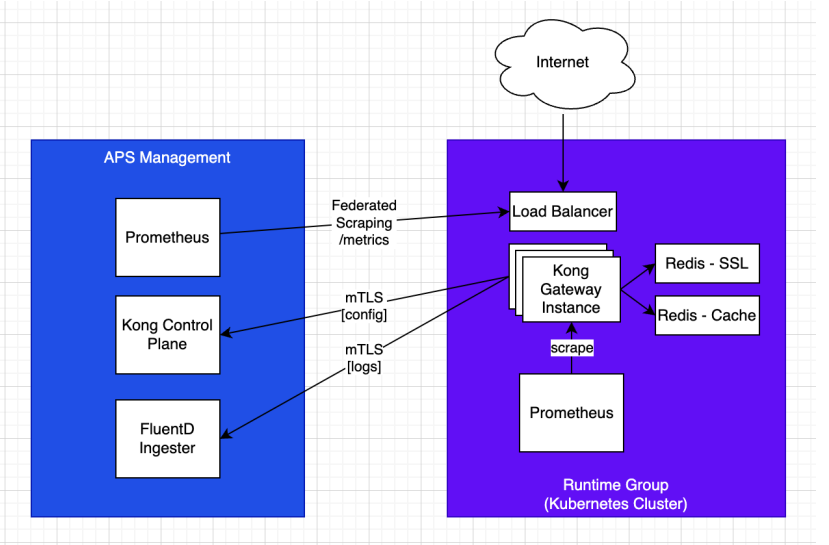
Kong supports the `ACME` protocol so it can be used to automatically issue/renew certificates through services such as Entrust and LetsEncrypt.

Deploy Kong Gateway

Deploy Kong to your own environment. An official Helm Chart is available to simplify deployment to a Kubernetes cluster.

What gets deployed?

- Kong Gateway
- Redis - SSL Certificate Storage and Memory Cache for Rate Limiting
- Prometheus - Metrics



Update DNS

For a typical cloud deployment to one of AWS, Azure or Google, either an IP or a unique domain name is provided as the entry point to Kong's Load Balancer.

This information is used to update DNS entries (either as an `A` record if its an IP or `CNAME` if its a domain name).

Configure your Services

Gateway Services can connect to any upstream service reachable within your environment - including serverless functions, containerized solutions or VM-based deployments.

Manage your APIs

All functionality that you enjoy from the API Services Portal is available for your new Client-Hosted Kong Gateway:

- Publish your API for discovery
- Issue credentials
- Approve access requests
- Administer gateway configuration
- View metrics

Demo

> Scenario: As an API Provider for the Ministry of Kittens and Puppies, I want to provision Kong's API Gateway to my Google Cloud environment so that I can use it to protect access to our sensitive kitten and puppy data.

Demo

1. Contact APS to request a new Runtime Group

ME "I want a new Runtime Group provisioned! APS, please provision a new Runtime Group."

APS Your new Runtime Group is provisioned - it's name is `clientdp-gcp-kong-proxy`. Your new TLS cert/key and CA certificate is attached. You will need this when you deploy your Kong Gateway.

```
runtime-group:
  name: 'clientdp-gcp-kong-proxy'
  namespace: 'clientdp-gcp'
  tls-node-crt: '<base64 cert>'
  tls-node-key: '<base64 key>'
  certificate-authority-data: '<base64 ca cert>'
```

Demo

2. Acquire a Domain Name and SSL certificate

I have purchased a new domain name for my ministry:

"ministryofpuppiesandkittens.site" from `namecheap.com`

Fortunately, my Ministry allows the use of LetsEncrypt for certificate issuing so we will use Kong to manage the SSL certificate creation and installation automatically.

Demo

3. Deploy Kong Gateway

> Wait! Kong Gateway deploys to a containerized environment - we will need to provision Google's EKS using Terraform templates provided by APS:

```
git clone https://github.com/bcgov/aps-client-hosted-kong-dp.git

cd aps-client-hosted-kong-dp/terraform/workspaces/gcp

gcloud auth login

terraform init
terraform apply
```

Demo

3. Deploy Kong Gateway

Deploy Kong Gateway using Kong's Konnect Quickstart Helm Chart:

```
git clone https://github.com/Kong/gcp-konnect-quickstart.git

cd gcp-konnect-quickstart

helm template "$APP_INSTANCE_NAME" chart/konnect-dp \
  --namespace "$NAMESPACE" \
  --set image.repository="$IMAGE_KONG_REPO" \
  --set image.tag="$TAG" \
  --set ca_crt="${CA_CRT}" \
  --set cluster_server_name="${SERVER_NAME}" \
  --set tls_key="$TLS_NODE_KEY" \
  --set tls_crt="$TLS_NODE_CRT" \
  --set proxy_tls_key="$TLS_CERTIFICATE_KEY" \
  --set proxy_tls_crt="$TLS_CERTIFICATE_CRT" \
  > "${APP_INSTANCE_NAME}_manifest.yaml"

kubectl apply -f "${APP_INSTANCE_NAME}_manifest.yaml" --namespace "${NAMESPACE}"
```

Demo

3. Deploy Kong Gateway

Deploy supporting services:

```
kubectl apply -f prom-rbac.yaml
helm upgrade --install prometheus -f prometheus.yaml prom/prometheus

helm upgrade \
  --install redis-ssl -f redis-ssl.yaml \
  --version 17.8.7 bitnami/redis

helm upgrade \
  --install redis -f redis.yaml \
  --version 16.12.1 bitnami/redis
```

Demo

4. Update DNS

After Kong has been deployed, a Load Balancer External IP will be available from the Kong Service.

Grab the `External IP` from Kong's Service:

```
export ZONE=us-central1
export CLUSTER=kong-dp-proxy-gcp-gke
gcloud container clusters get-credentials "$CLUSTER" --zone "$ZONE"

kubectl get services
```

Go to your DNS Registry `namecheap.com` and update the `A` record to reflect the IP.

Demo

5. Configure your Services

I have deployed a private `Cloud Function` `us-central1-kong-dp-proxy-gcp.cloudfunctions.net` that I want to protect with Kong.

I first configure the service:

```
services:
- name: a-service-for-clientdp-gcp
  tags: [ns.clientdp-gcp]
  host: us-central1-kong-dp-proxy-gcp.cloudfunctions.net
  path: /function-1
  port: 443
  protocol: https
  routes:
    - name: a-service-for-clientdp-gcp-route
      tags: [ns.clientdp-gcp]
      hosts:
        - api.ministryofpuppiesandkittens.site
      paths:
        - /sample$
      methods:
        - GET
      strip_path: true
```

Demo

5. Configure your Services

Run `gwa pg` to publish the Gateway configuration.

Verify by running:

```
curl -v "https://api.ministryofpuppiesandkittens.site/sample"
```

A good response is a simple `all my kitten and puppy data`.

> NOTE: If `ACME` was enabled on Kong, then the first time you call the endpoint, Kong will initiate SSL issuing if the SSL certificate is new. Wait a minute before trying again and you should receive a valid response.

Demo

6. Manage your APIs

Serverless Functions	https://console.cloud.google.com/functions/list?project=kong-dp-proxy-gcp
Kubernetes Cluster	https://console.cloud.google.com/kubernetes/list/overview?project=kong-dp-proxy-gcp
Kong Gateway	https://console.cloud.google.com/kubernetes/deployment/us-central1/kong-dp-proxy-gcp-gke/default/aps-kong-gateway-gcp-kong/overview?project=kong-dp-proxy-gcp
API Services Portal	https://api-gov-bc-ca.dev.api.gov.bc.ca/manager/services
APS Operations	https://ops-grafana-dev-b8840c-dev.apps.gold.devops.gov.bc.ca
Test Load	https://console.apps.gold.devops.gov.bc.ca/k8s/ns/b8840c-tools/deployments/test-load-gcp-dp-generic-api

Conclusion

Adding the capability for teams to deploy a Kong Gateway within their own infrastructure is a great way for ministries with strong requirements around data sensitivity, low latency and infrastructure autonomy, to offload the full suite of API Management services to APS while keeping the critical Gateway traffic and data within their control.

If interested in learning more, please reach out to us!