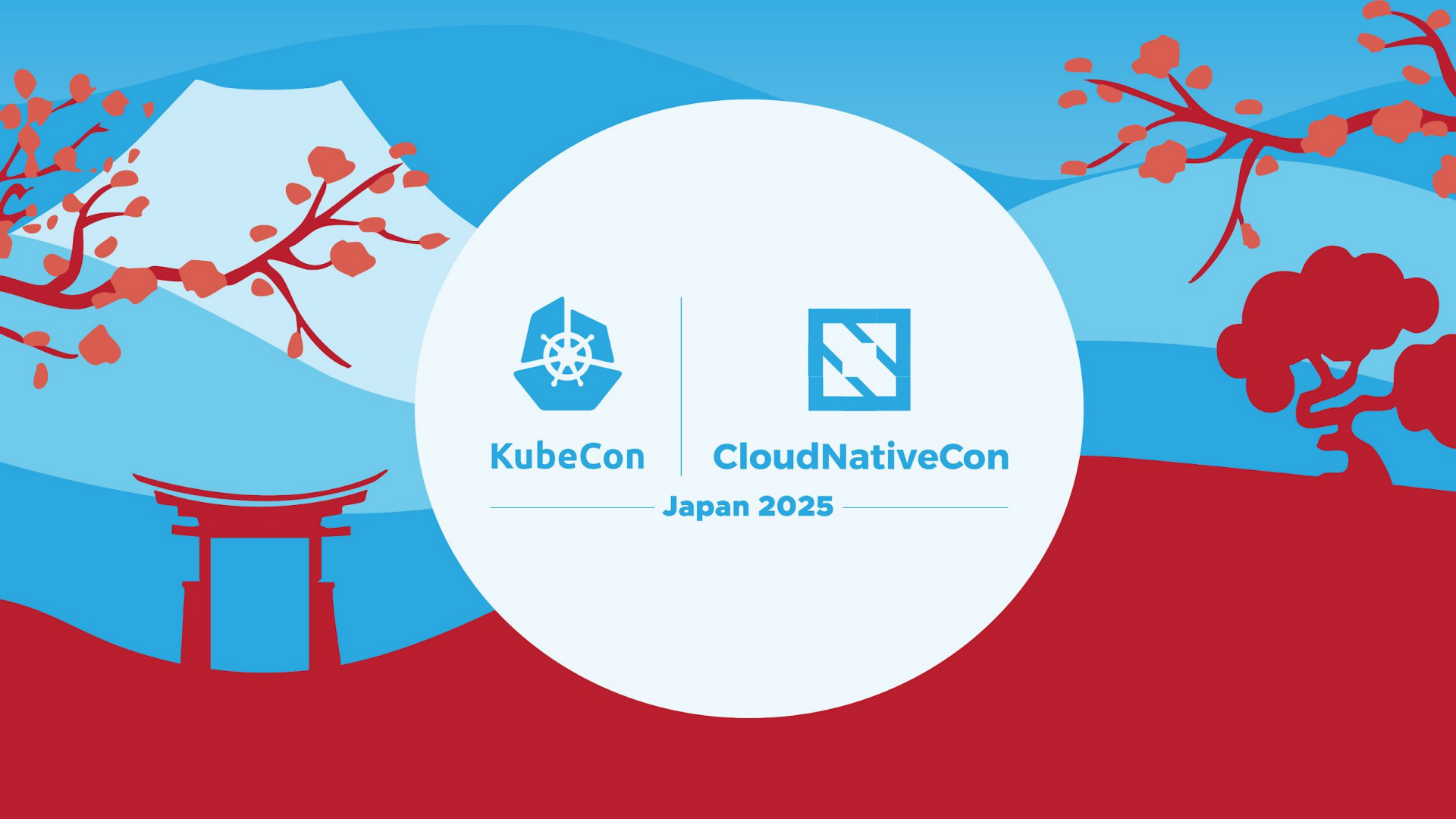

KubeCon
CloudNativeCon
Japan 2025

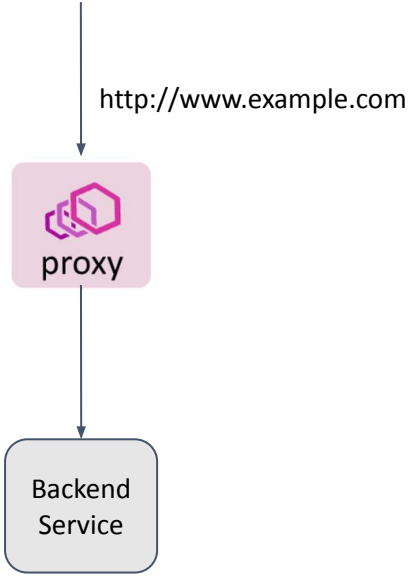
Envoy Gateway Policies:

Unlocking the Full Power of Envoy for API Gateways!

Envoy Gateway Maintainer | Envoy Contributor | CNCF Ambassador
Huabing (Robin) Zhao@Tetrate

Me Trying to Learn Skateboard Tricks on My Own

Envoy Configuration Can Be… a Lot



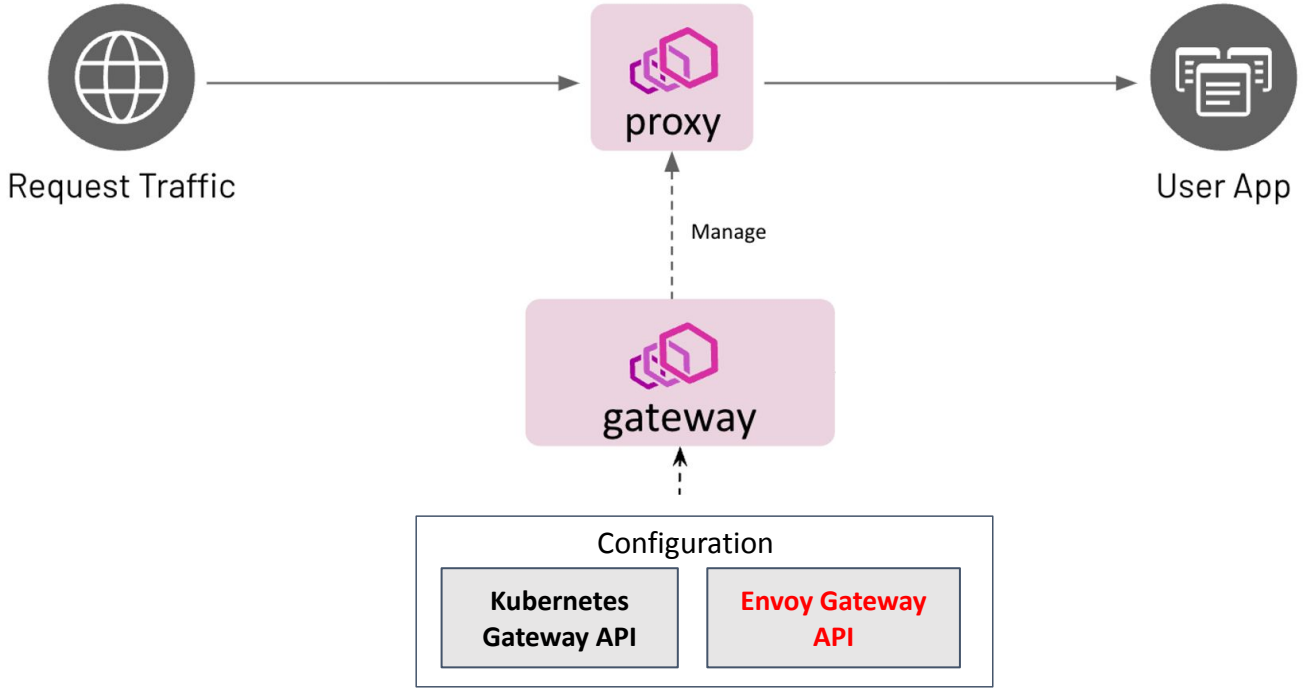

```
envoy-gateway-system:
  envoy-default-eg-e41e7b31-5c446cd7b7-77t9q:
                setNodeOnFirstMessageOnly: true
                transportApiVersion: V3
                      name: envoy.listener.http_inspector
                    name: envoy.matching.inputs.destination_ip
                    typeUrls:
                    - envoy.extensions.matching.common_inputs.network.v3.DestinationIPI
                              namespace: default
                        name: httproute/default/myapp/rule/0/match/0/www_example_com
                        route:
                          cluster: httproute/default/myapp/rule/0
                          upgradeConfigs:
                          - upgradeType: websocket
    versionInfo: b1d7046b2f4f54b67a809fab4e3e32a82e91ef1de9dd293d12703ceff0
    - lastUpdated: "2025-05-15T11:22:07.949Z"
      routeConfig:
        '@type': type.googleapis.com/envoy.config.route.v3.RouteConfiguration
        ignorePortInHostMatching: true
        name: default/eg/https
        virtualHosts:
        - domains:
          - www.example.com
          metadata:
            filterMetadata:
              envoy-gateway:
                resources:
                - kind: Gateway
                  name: eg
                  namespace: default
                  sectionName: https
          name: default/eg/https/www_example_com
          routes:
          - match:
              pathSeparatedPrefix: /myapp
            metadata:
              filterMetadata:
                envoy-gateway:
                  resources:
                  - kind: HTTPRoute
                    name: myapp
                    namespace: default
            name: httproute/default/myapp/rule/0/match/0/www_example_com
            route:
              cluster: httproute/default/myapp/rule/0
              upgradeConfigs:
              - upgradeType: websocket
    versionInfo: c63a64cd6d88768f6bec76f72628cf412ea49f57062297890f47bcda98
  staticRouteConfigs:
  - lastUpdated: "2025-05-15T11:18:33.891Z"
    routeConfig:
```

Envoy is a Powerful HTTP Proxy, but …

My take: trying to configure Envoy as an API Gateway without the right tools is like learning to skateboard without a coach — risky and painful.

Envoy Gateway: The Easy Way to Run Envoy

- **Envoy Gateway** is an **open source** project for managing **Envoy Proxy** as a Kubernetes-based or standalone **API Gateway**.
- **Kubernetes Gateway API** resources are used to automatically provision and configure the managed Envoy Proxies.



Gateway API: GatewayClass

GatewayClass: defines a class of Gateways that share a common configuration and behaviour

Infrastructure provider

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: eg
spec:
  controllerName: gateway.envoyproxy.io/gatewayclass-controller
  parametersRef:
    group: gateway.envoyproxy.io
    kind: EnvoyProxy
    name: config
    namespace: envoy-gateway-system
```

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: EnvoyProxy
metadata:
  namespace: envoy-gateway-system
  name: config
spec:
  telemetry:
    tracing:
      samplingRate: 100
      provider:
        host: otel-collector.monitoring.svc.cluster.local
        port: 4317
        type: OpenTelemetry
```

Gateway API: Gateway

Gateway: specifies how traffic can enter the cluster

Cluster operator


```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: foo-gateway
spec:
  gatewayClassName: eg
  listeners:
    - name: https
      protocol: HTTPS
      hostname: "foo.example.com"
      port: 443
      tls:
        mode: Terminate
        certificateRefs:
          - name: server-cert
```

Gateway API: HTTPRoute

HTTPRoute: defines rules for mapping requests from a Gateway to Backend Services.

Site Developer →

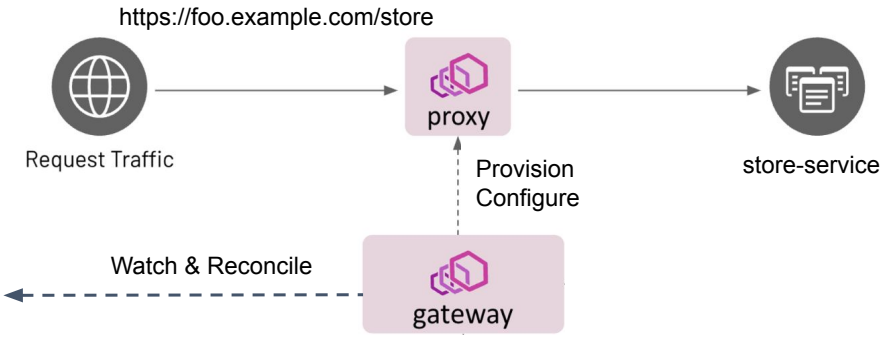
```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: store-route
spec:
  parentRefs:
    - name: foo-gateway
  hostnames:
    - "foo.example.com"
  rules:
    - matches:
      - path:
          value: "/store"
      backendRefs:
        - kind: Service
          name: store-service
          port: 3000
```

How These Pieces Come Together

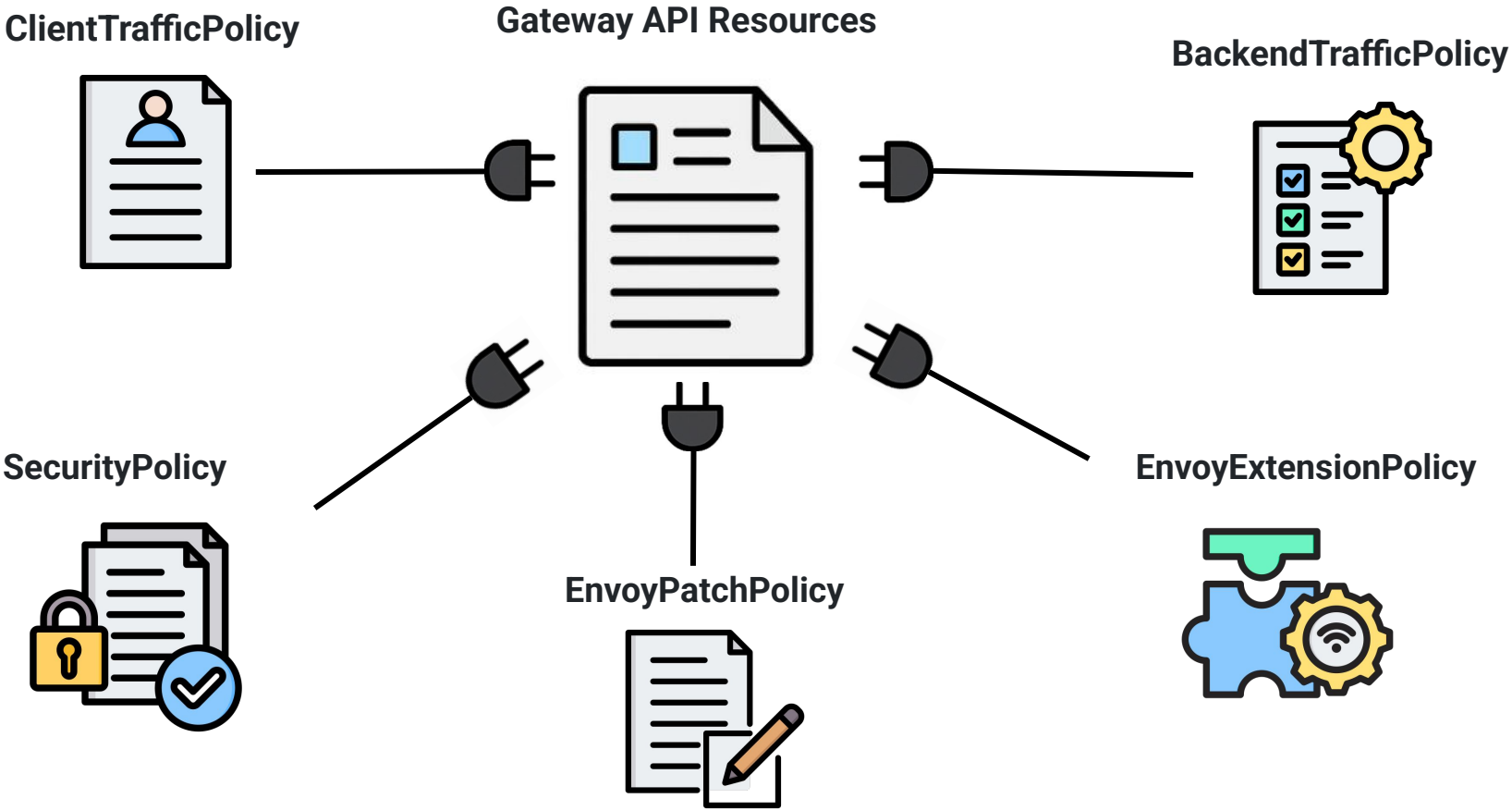

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: eg
spec:
  controllerName: gateway.envoyproxy.io/gatewayclass-controller
  parametersRef:
    group: gateway.envoyproxy.io
    kind: EnvoyProxy
    name: config
    namespace: envoy-gateway-system
```

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: foo-gateway
spec:
  gatewayClassName: eg
  listeners:
    - name: https
      protocol: HTTPS
      hostname: "foo.example.com"
      port: 443
      tls:
        mode: Terminate
        certificateRefs:
          - name: server-cert
```

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: store-route
spec:
  parentRefs:
    - name: foo-gateway
  hostnames:
    - "foo.example.com"
  rules:
    - matches:
      - path:
          value: "/store"
      backendRefs:
        - kind: Service
          name: store-service
          port: 3000
```



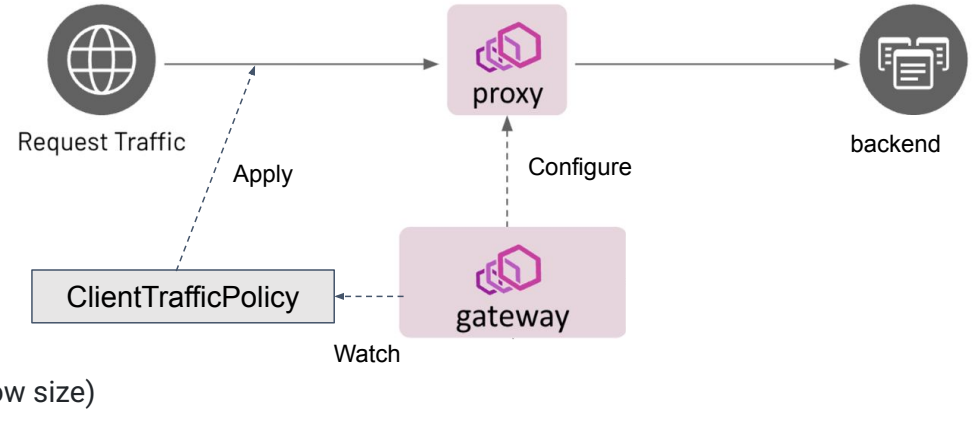
Envoy Gateway Policies: Unlocking the Full Power of Envoy



ClientTrafficPolicy

Traffic policy for client connections (connections between client and Envoy)

- TCP settings for downstream client connections
  - TCP Keepalive
  - TCP Timeout (TCP Idle time)
  - Connection Limit
  - Socket and Connection Buffer size
- TLS settings for downstream client connections
  - Should and how to verify the client cert
  - TLS options: version, ciphers, ALPN, etc.
- HTTP settings for downstream client connections
  - HTTP Timeout (Request timeout, HTTP Idle time)
  - HTTP1/HTTP2/HTTP3 settings (e.g.,: HTTP2 stream window size)
- Other downstream client connections-related configurations
  - Whether Proxy protocol is enabled or not on the client connection
  - How to detect the original client IP of the client request



Please note: not all features can be applied to all Listener types.

If a targeted Listener does not support a feature, the feature will be ignored. For example, the HTTP2 setting will be ignored if the Listener is a TCP Listener.

ClientTrafficPolicy

Targets

- Gateway: ClientTrafficPolicy applies on all Listeners to the targeted Gateway
- Listener: ClientTrafficPolicy applies on the specified Listener only

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: eg
spec:
  gatewayClassName: internet
  listeners:
    - name: http
      protocol: HTTP
      port: 80
    - name: https
      protocol: HTTPS
      port: 443
      tls:
        mode: Terminate
        certificateRefs:
          - name: server-cert
```

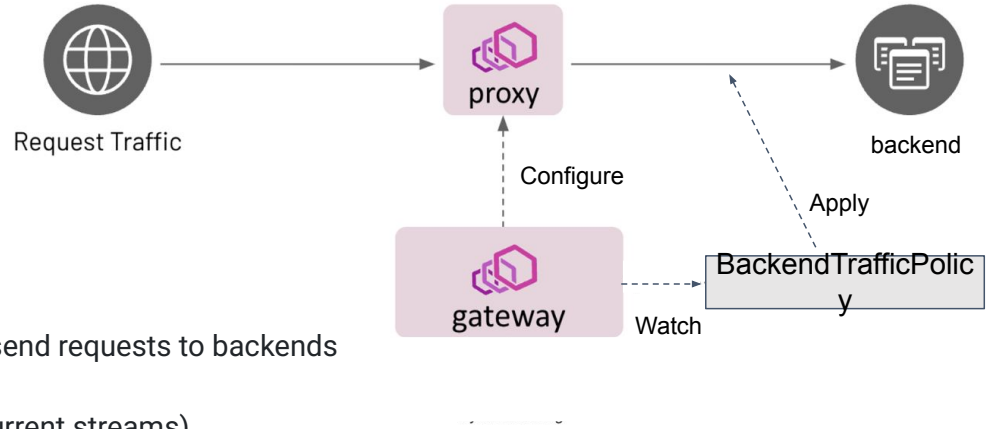
```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: ClientTrafficPolicy
metadata:
  name: client-traffic-policy-gateway
spec:
  targetRefs:
    - group: gateway.networking.k8s.io
      kind: Gateway
      name: eg
  tcpKeepalive:
    idleTime: 20m
    interval: 60s
    probes: 3
  connection:
    bufferLimit: 16Ki
  clientIPDetection:
    xForwardedFor:
      numTrustedHops: 2
  timeout:
    http:
      requestReceivedTimeout: 2s
      idleTimeout: 5s
```

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: ClientTrafficPolicy
metadata:
  name: client-traffic-policy-https-listener
spec:
  targetRefs:
    - group: gateway.networking.k8s.io
      kind: Gateway
      name: eg
      sectionName: https
  tcpKeepalive:
    idleTime: 20m
    interval: 60s
    probes: 3
  connection:
    bufferLimit: 16Ki
  clientIPDetection:
    xForwardedFor:
      numTrustedHops: 2
  timeout:
    http:
      requestReceivedTimeout: 2s
      idleTimeout: 5s
  tls:
    clientValidation:
      caCertificateRefs:
        - name: client-ca
```

BackendTrafficPolicy

Traffic policy for backend connections (connections between Envoy and backend)

- Global and Local RateLimit
- Retries
- Load Balancing
  - Algorithms: ConsistentHash, LeastRequest, Random, RoundRobin
  - Support SlowStart to gradually warm up JAVA applications
- Circuit Breaker
  - Max connections/requests per connection
  - Max pending requests/parallel requests/parallel retries
- TCP Keepalive
- Socket and Connection Buffer size
- TCP and HTTP Timeout
- Active and Passive Health Check (Outlier Detection)
- Enable Proxy protocol when communicating with the backend
- Use the same HTTP protocol that the incoming request used to send requests to backends
- DNS refresh rate and TTL for DNS type backend cluster
- HTTP2 settings (e.g., HTTP2 stream window size and max concurrent streams)



Please note: not all features can be applied to all xRoute types. If a targeted xRoute does not support a feature, the feature will be ignored. For example, the RateLimit setting will be ignored if the Route is a TCP Route.

BackendTrafficPolicy

Targets

- Gateway: BackendTrafficPolicy applies on all xRoute to the targeted Gateway
- xRoute: BackendTrafficPolicy applies on the specified xRoute only

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: eg
spec:
  gatewayClassName: internet
  listeners:
    - name: http
      protocol: HTTP
      port: 80
```

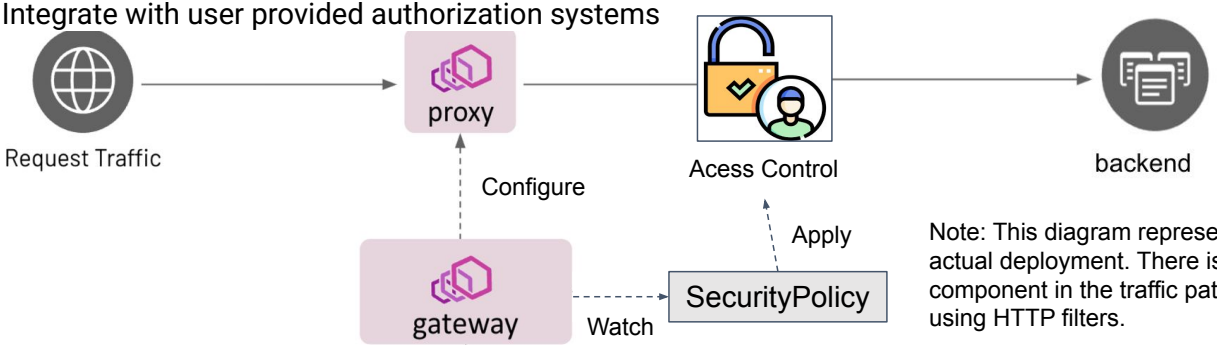
```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: http-route
spec:
  parentRefs:
    - name: eg
  hostnames:
    - "foo.bar.com"
  rules:
    - backendRefs:
        - name: foo-svc
          port: 8080
```

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: BackendTrafficPolicy
metadata:
  name: backend-traffic-policy-http-route
spec:
  targetRefs:
  - group: gateway.networking.k8s.io
    kind: HTTPRoute
    name: http-route
  rateLimit:
    type: Global
    global:
      rules:
      - clientSelectors:
        - headers:
          - type: Distinct
            name: x-user-id
        limit:
          requests: 100
          unit: Second
    loadBalancer:
      type: ConsistentHash
      consistentHash:
        type: SourceIP
    circuitBreaker:
      maxPendingRequests: 1024
      maxParallelRequests: 1024
```

SecurityPolicy

Security Settings for Gateway (Access Control, Authentication, and Authorization Policies)

- CORS: Access control based on the origin of the request
- Authentication
  - HTTP Basic Auth
  - API Key Auth
  - OIDC
    - Integrate with any IdPs: Google, Auth0, Azure AD, Keycloak, Okta, etc.
    - Support SSO across multiple applications
  - JWT Auth
- Authorization
  - Principal: Original request IP, JWT Claims, Basic Auth user, etc.
  - Action: allow/deny access to targeted HTTP Routes
- Ext Auth: Integrate with user provided authorization systems



Note: This diagram represents a logical structure rather than an actual deployment. There is no standalone ‘Access Control’ component in the traffic path; it is implemented as part of the Envoy using HTTP filters.

SecurityPolicy

Targets

- Gateway: SecurityPolicy applies on all xRoute on the targeted Gateway
- xRoute: SecurityPolicy applies on the specified xRoute only

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: eg
spec:
  gatewayClassName: internet
  listeners:
    - name: http
      protocol: HTTP
      port: 80
```

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: http-route
spec:
  parentRefs:
    - name: eg
  hostnames:
    - "foo.bar.com"
  rules:
    - backendRefs:
        - name: foo-svc
          port: 8080
```

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: SecurityPolicy
metadata:
  name: scurity-policy-http-route
spec:
  targetRefs:
    - group: gateway.networking.k8s.io
      kind: HTTPRoute
      name: http-route
  oidc:
    provider:
      issuer: "https://accounts.google.com"
    clientID: "client1.apps.googleusercontent.com"
    clientSecret:
      name: "my-app-client-secret"
    redirectURL: "https://www.example.com/myapp/oauth2/callback"
    logoutPath: "/myapp/logout"
  authorization:
    defaultAction: Deny
    rules:
    - action: Allow
      principal:
        clientCIDRs:
        - 10.0.1.0/24
```

EnvoyExtensionPolicy

Expand Envoy's functionality with custom extensions. Envoy currently supports:

- **Lua Script**
  - Simple, inline scripting for custom logic
  - Great for lightweight request/response tweaks
  - Easiest to use — no build or packaging needed
- **WASM (WebAssembly)**
  - Runs inside Envoy for high performance
  - Supports OCI image & HTTP-based deployment
  - Versioned, access-controlled, fewer moving parts
- **External Process**
  - Runs outside Envoy for full isolation
  - Uses any gRPC-capable language
  - Scales independently, avoids crashing Envoy
  - Slightly heavier due to network calls and deployment

EnvoyExtensionPolicy - Lua

Inline Lua script

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: EnvoyExtensionPolicy
metadata:
  name: example-lua-2
  namespace: gateway-conformance-infra
spec:
  targetRefs:
    - group: gateway.networking.k8s.io
      kind: HTTPRoute
      name: example-route-2-with-lua
  lua:
    - type: Inline
      inline: |
        function envoy_on_response(response_handle)
          response_handle:headers():add("X-Custom-Lua-Header", "lua_value_2")
          response_handle:headers():add("X-Custom-Lua-Another-Header", "lua_another_value")
        end
```

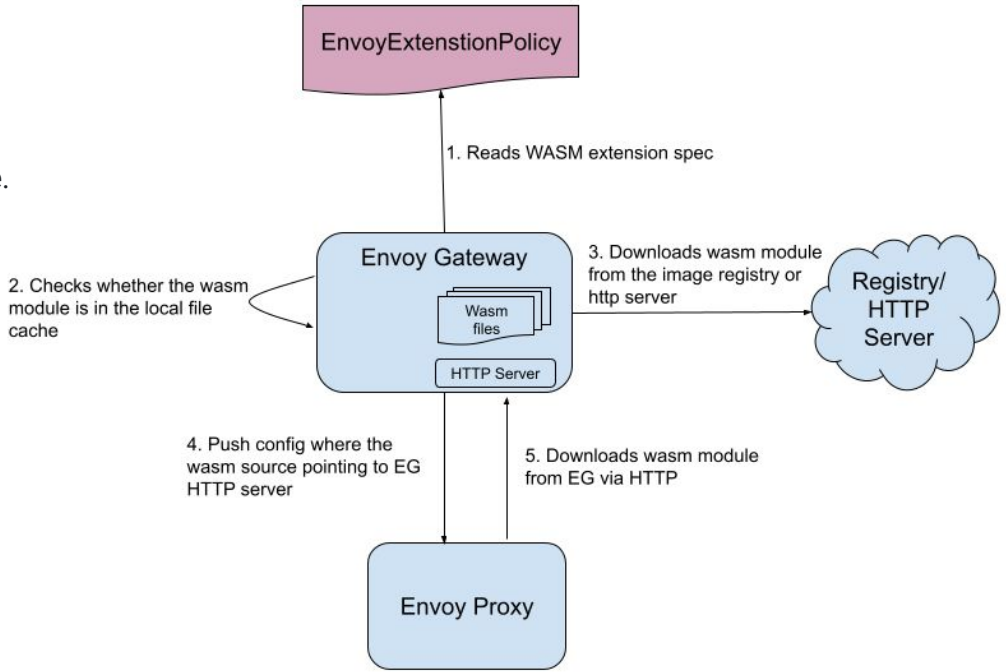
Lua script in ConfigMap

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: EnvoyExtensionPolicy
metadata:
  name: example-lua-1
  namespace: gateway-conformance-infra
spec:
  targetRefs:
    - group: gateway.networking.k8s.io
      kind: HTTPRoute
      name: example-route-1-with-lua
  lua:
    - type: ValueRef
      valueRef:
        name: cm-example-lua
        kind: ConfigMap
        group: v1
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: cm-example-lua
  namespace: gateway-conformance-infra
data:
  lua: |
    function envoy_on_response(response_handle)
      response_handle:headers():add("X-Custom-Lua-Header", "lua_value_1")
    end
```

EnvoyExtensionPolicy - Wasm

Envoy Gateway support Wasm OCI image as a remote Wasm code source.

- **Versioning**: Users can use the tag feature of the OCI image to manage the version of the Wasm module.
- **Security**: Users can use private registries to store the Wasm module.
- **Distribution**: Users can use the existing distribution mechanism of the OCI registry to distribute the Wasm module.



EnvoyExtensionPolicy - Wasm

KubeCon CloudNativeCon Japan 2025

OCI Image Wasm source

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: EnvoyExtensionPolicy
metadata:
  name: wasm-test
spec:
  targetRefs:
  - group: gateway.networking.k8s.io
    kind: HTTPRoute
    name: backend
  wasm:
  - name: wasm-filter-1
    rootID: my_root_id
    code:
      type: Image
      image:
        url: zhaohuabing/testwasm:v0.0.1
  - name: wasm-filter-2
    rootID: "my-root-id"
    code:
      type: Image
      image:
        url: oci://my.private.regisgtry/wasm-filter-2:v1.0.0
        pullSecretRef:
          name: my-pull-secret
        sha256: a1efca12ea51069abb123bf9c77889fcc2a31cc5483
    config:
      parameter1: value1
      parameter2: value2
```

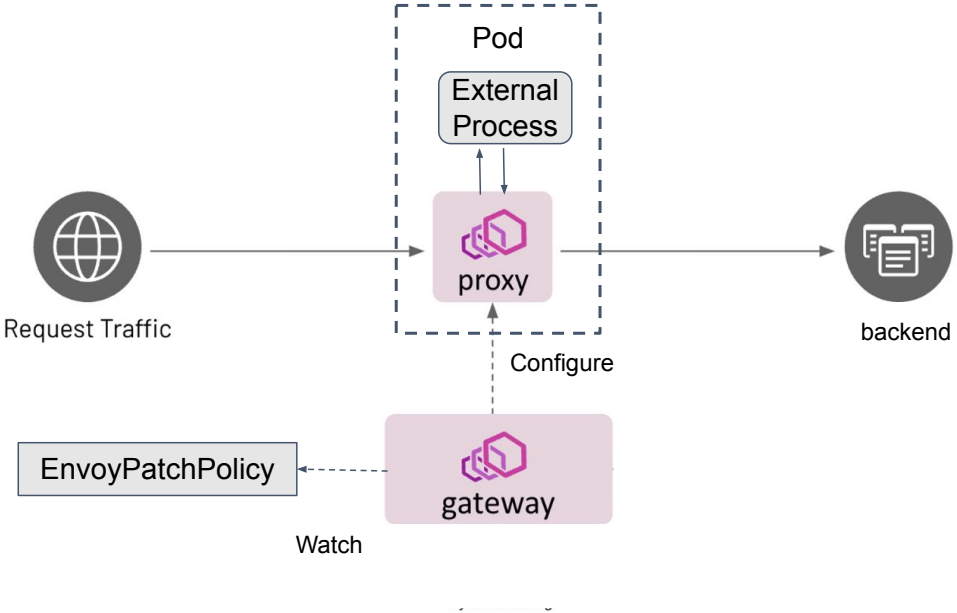
HTTP Wasm source

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: EnvoyExtensionPolicy
metadata:
  name: wasm-test
spec:
  targetRefs:
  - group: gateway.networking.k8s.io
    kind: HTTPRoute
    name: backend
  wasm:
  - name: wasm-filter-1
    code:
      type: HTTP
      http:
        url: https://www.example.com/wasm-filter-1.wasm
        sha256: 746df05c8f3a0b07a46c0967cfbc5cbe5b9d48d0f79
    config:
      parameter1:
        key1: value1
        key2: value2
      parameter2: value3
```

EnvoyExtensionPolicy - External Process

External Process Extension

- The External Process can be deployed as a sidecar to minimize network latency.
- A Unix Domain Socket (UDS) type of Backend API can be used to reference the sidecar service.




```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: EnvoyExtensionPolicy
metadata:
  namespace: default
  name: policy-for-http-route
spec:
  targetRef:
    group: gateway.networking.k8s.io
    kind: HTTPRoute
    name: httproute-1
  extProc:
  - backendRefs:
    - Name: uds-ext-proc
      kind: Backend
      group: gateway.envoyproxy.io
```

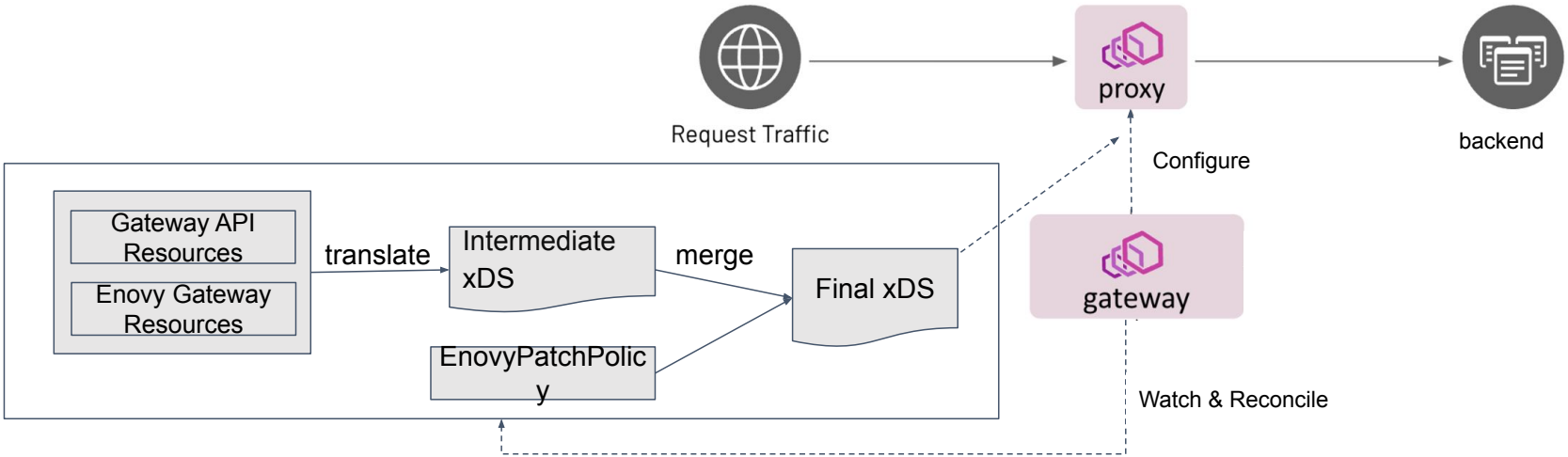
```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: Backend
metadata:
  name: uds-ext-proc
spec:
  endpoints:
  - unix:
      path: /var/run/ext-proc/extproc.sock
```

EnvoyPatchPolicy

Add arbitrary patches to the generated xDS; this is especially useful for:

- Verifying prototype before formally landing a feature to EG.
- Implementing temporary workarounds for features not yet supported by EG.

Caveat: The compatibility of EnvoyPatchPolicy is not guaranteed. An EnvoyPatchPolicy that functions with a specific version may not work following an upgrade due to changes in the xDS translation. Please consider this when using EnvoyPatchPolicy.



EnvoyPatchPolicy

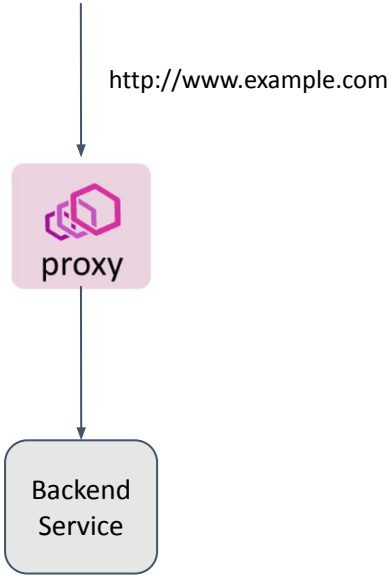
Enable EnvoyPatchPolicy

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: envoy-gateway-config
  namespace: envoy-gateway-system
data:
  envoy-gateway.yaml: |
    apiVersion: gateway.envoyproxy.io/v1alpha1
    kind: EnvoyGateway
    provider:
      type: Kubernetes
    gateway:
      controllerName: gateway.envoyproxy.io/gatewayclass-controller
    extensionApis:
      enableEnvoyPatchPolicy: true
```

EnvoyPatchPolicy for custom response

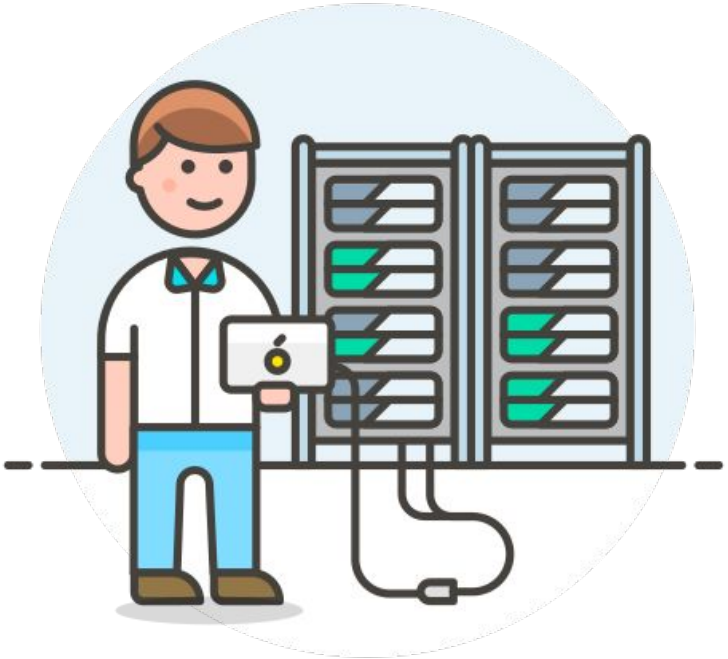
```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: EnvoyPatchPolicy
metadata:
  name: custom-response-patch-policy
  namespace: default
spec:
  targetRef:
    group: gateway.networking.k8s.io
    kind: Gateway
    name: eg
  type: JSONPatch
  jsonPatches:
    - type: "type.googleapis.com/envoy.config.listener.v3.Listener"
      # The listener name is of the form <GatewayNamespace>/<GatewayName>/<GatewayListenerName>
      name: default/eg/http
      operation:
        op: add
        path: "/default_filter_chain/filters/0/typed_config/local_reply_config"
        value:
          mappers:
          - filter:
              status_code_filter:
                comparison:
                  op: EQ
                  value:
                    default_value: 404
                    runtime_key: key_b
            status_code: 406
            body:
              inline_string: "could not find what you are looking for"
```

Managing Envoy with Envoy Gateway




```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: eg
spec:
  controllerName: gateway.envoyproxy.io/gatewayclass-controller
---
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: eg
spec:
  gatewayClassName: eg
  listeners:
  - allowedRoutes:
      namespaces:
        from: Same
    name: http
    port: 80
    protocol: HTTP
---
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: myapp
spec:
  parentRefs:
  - name: eg
  hostnames: ["www.example.com"]
  rules:
  - matches:
    - path:
        type: PathPrefix
        value: /myapp
    backendRefs:
    - name: backend
      port: 3000
```

Demo: SecurityPolicy



SecurityPolicy

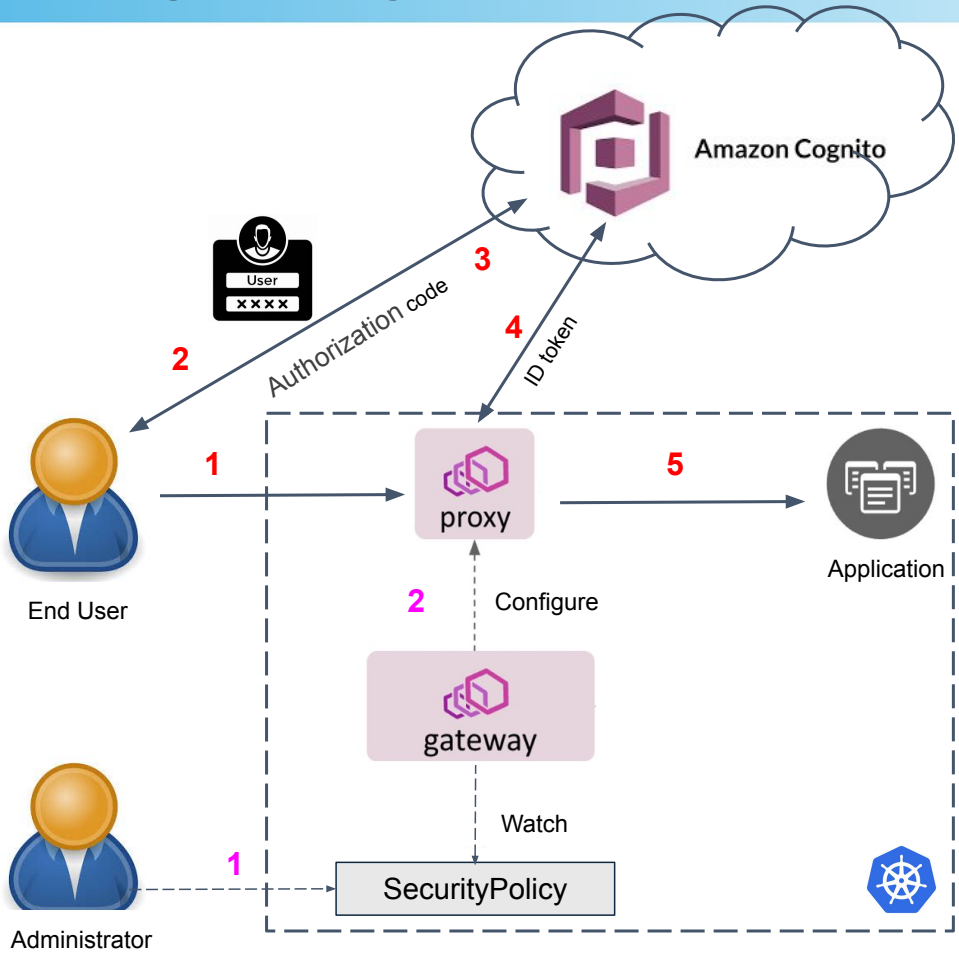
Questions?

- Envoy Gateway simplifies Envoy
- Built on the Gateway API
- Extended for full Envoy capability

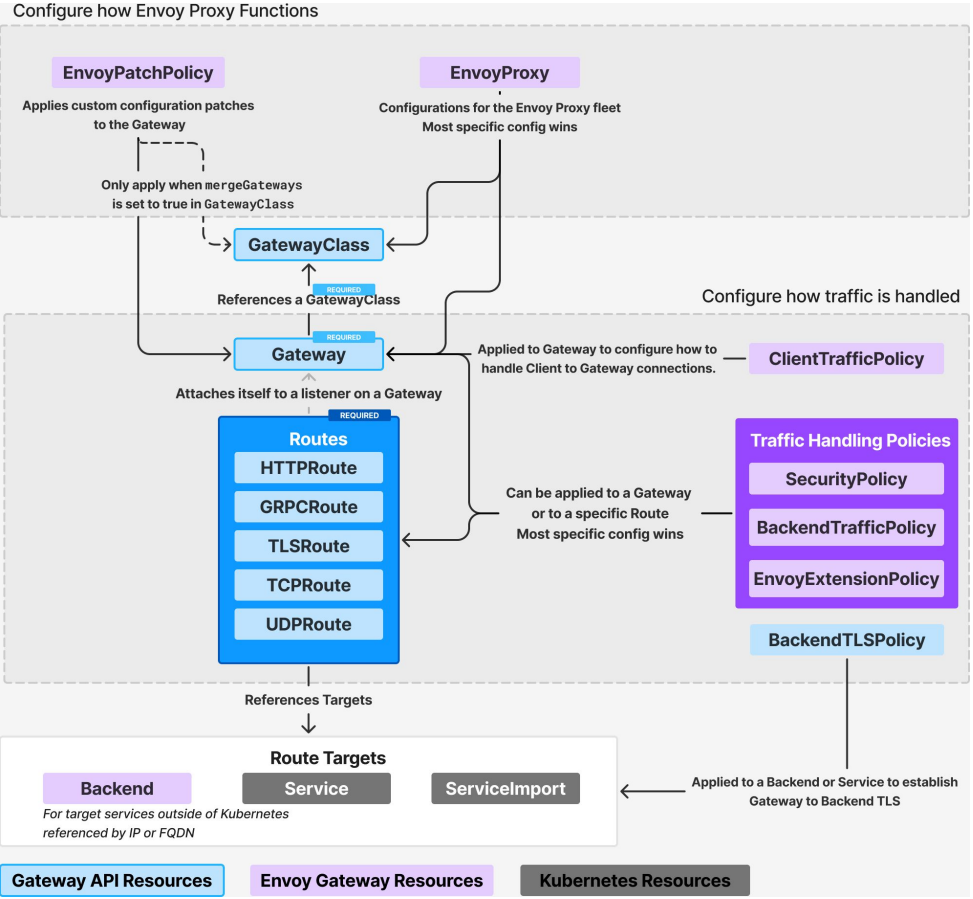
Docs: https://gateway.envoyproxy.io/
GitHub: https://github.com/envoyproxy/gateway
Join us in the envoy **#gateway-users** slack channel

Backup Slides

Demo: SecurityPolicy - OIDC



Envoy Gateway Extensions



Envoy Gateway API Resources

- **EnvoyProxy:** Represents the deployment and configuration of the Envoy proxy within a Kubernetes cluster, managing its lifecycle and settings.
- **Backend:** A resource that makes routing to cluster-external backends easier and makes access to external processes via Unix Domain Sockets possible.
- **xPolicy :** Additional policies to enhance Gateway API resources
  - **ClientTrafficPolicy: :** Configuration for downstream client to Envoy listener.
  - **BackendTrafficPolicy :** Configuration for Envoy to backend service.
  - **SecurityPolicy :** Configuration for security settings.
  - **EnvoyExtensionPolicy :** Configuration for Enovy extensions(Wasm, ExtProc).
  - **EnvoyPatchPolicy :** Abritary patches to the generated xDS.