Envoy Example

Let’s take the Web frontend and customer service as an example and see how Envoy determines where to send the request from the web frontend to the customer service (customers.default.svc.cluster.local). Using the istioctl proxy-config command, we can list all listeners of the web frontend pod.

$ istioctl proxy-config listeners web-frontend-64455cd4c6-p6ft2

ADDRESS PORT MATCH DESTINATION

10.124.0.10 53 ALL Cluster: outbound|53||kube-dns.kube-system.svc.cluster.local

0.0.0.0 80 ALL PassthroughCluster

10.124.0.1 443 ALL Cluster: outbound|443||kubernetes.default.svc.cluster.local

10.124.3.113 443 ALL Cluster: outbound|443||istiod.istio-system.svc.cluster.local

10.124.7.154 443 ALL Cluster: outbound|443||metrics-server.kube-system.svc.cluster.local

10.124.7.237 443 ALL Cluster: outbound|443||istio-egressgateway.istio-system.svc.cluster.local

10.124.8.250 443 ALL Cluster: outbound|443||istio-ingressgateway.istio-system.svc.cluster.local

10.124.3.113 853 ALL Cluster: outbound|853||istiod.istio-system.svc.cluster.local

0.0.0.0 8383 ALL PassthroughCluster

0.0.0.0 15001 ALL PassthroughCluster

0.0.0.0 15006 ALL Inline Route: /\*

0.0.0.0 15010 ALL PassthroughCluster

10.124.3.113 15012 ALL Cluster: outbound|15012||istiod.istio-system.svc.cluster.local

0.0.0.0 15014 ALL PassthroughCluster

0.0.0.0 15021 ALL Non-HTTP/Non-TCP

10.124.8.250 15021 ALL Cluster: outbound|15021||istio-ingressgateway.istio-system.svc.cluster.local

0.0.0.0 15090 ALL Non-HTTP/Non-TCP

10.124.7.237 15443 ALL Cluster: outbound|15443||istio-egressgateway.istio-system.svc.cluster.local

10.124.8.250 15443 ALL Cluster: outbound|15443||istio-ingressgateway.istio-system.svc.cluster.local

10.124.8.250 31400 ALL Cluster: outbound|31400||istio-ingressgateway.istio-system.svc.cluster.local

The request from the web frontend to customers is an outbound HTTP request to port 80. This means that it gets handed off to the **0.0.0.0:80** virtual listener. We can use Istio CLI to filter the listeners by address and port. You can add the -o json to get a JSON representation of the listener:

$ istioctl proxy-config listeners web-frontend-58d497b6f8-lwqkg --address 0.0.0.0 --port 80 -o json

...

"rds": {

"configSource": {

"ads": {},

"resourceApiVersion": "V3"

},

"routeConfigName": "80"

},

...

The listener uses RDS (Route Discovery Service) to find the route configuration (80 in our case). Routes are attached to listeners and contain rules that map **virtual hosts** to clusters. This allows us to create traffic routing rules because Envoy can look at headers or paths (the request metadata) and route traffic.

A route selects a **cluster**. A cluster is a group of similar upstream hosts that accept traffic - it’s a collection of endpoints. For example, the collection of all instances of the Web Frontend service is a cluster. We can configure resiliency features within a cluster, such as circuit breakers, outlier detection, and TLS config.

Using the routes command, we can get the route details by filtering all routes by the name:

$ istioctl proxy-config routes web-frontend-58d497b6f8-lwqkg --name 80 -o json

[

{

"name": "80",

"virtualHosts": [

{

"name": "customers.default.svc.cluster.local:80",

"domains": [

"customers.default.svc.cluster.local",

"customers.default.svc.cluster.local:80",

"customers",

"customers:80",

"customers.default.svc.cluster",

"customers.default.svc.cluster:80",

"customers.default.svc",

"customers.default.svc:80",

"customers.default",

"customers.default:80",

"10.124.4.23",

"10.124.4.23:80"

],

],

"routes": [

{

"match": {

"prefix": "/"

},

"route": {

"cluster": "outbound|80|v1|customers.default.svc.cluster.local",

"timeout": "0s",

"retryPolicy": {

"retryOn": "connect-failure,refused-stream,unavailable,cancelled,retriable-status-codes",

"numRetries": 2,

"retryHostPredicate": [

{

"name": "envoy.retry\_host\_predicates.previous\_hosts"

}

],

"hostSelectionRetryMaxAttempts": "5",

"retriableStatusCodes": [

503

]

},

"maxGrpcTimeout": "0s"

},

...

The route 80 configuration has a virtual host for each service. However, because our request is being sent to customers.default.svc.cluster.local, Envoy selects the virtual host (customers.default.svc.cluster.local:80) that matches one of the domains.

Once the domain is matched, Envoy looks at the routes and picks the first one that matches the request. Since we don’t have any special routing rules defined, it matches the first (and only) route that’s defined and instructs Envoy to send the request to the cluster named outbound|80|v1|customers.default.svc.cluster.local.

Note the v1 in the cluster’s name is because we have a DestinationRule deployed that creates the v1 subset. If there are no subsets for a service, that part if left blank: outbound|80||customers.default.svc.cluster.local.

Now that we have the cluster name, we can look up more details. To get an output that clearly shows the FQDN, port, subset and other information, you can omit the -o json flag:

$ istioctl proxy-config cluster web-frontend-58d497b6f8-lwqkg --fqdn customers.default.svc.cluster.local

SERVICE FQDN PORT SUBSET DIRECTION TYPE DESTINATION RULE

customers.default.svc.cluster.local 80 - outbound EDS customers.default

customers.default.svc.cluster.local 80 v1 outbound EDS customers.default

Finally, using the cluster name, we can look up the actual endpoints the request will end up at:

$ istioctl proxy-config endpoints web-frontend-58d497b6f8-lwqkg --cluster "outbound|80|v1|customers.default.svc.cluster.local"

ENDPOINT STATUS OUTLIER CHECK CLUSTER

10.120.0.4:3000 HEALTHY OK outbound|80|v1|customers.default.svc.cluster.local

The endpoint address equals the pod IP where the Customer application is running. If we scale the customers deployment, additional endpoints show up in the output, like this:

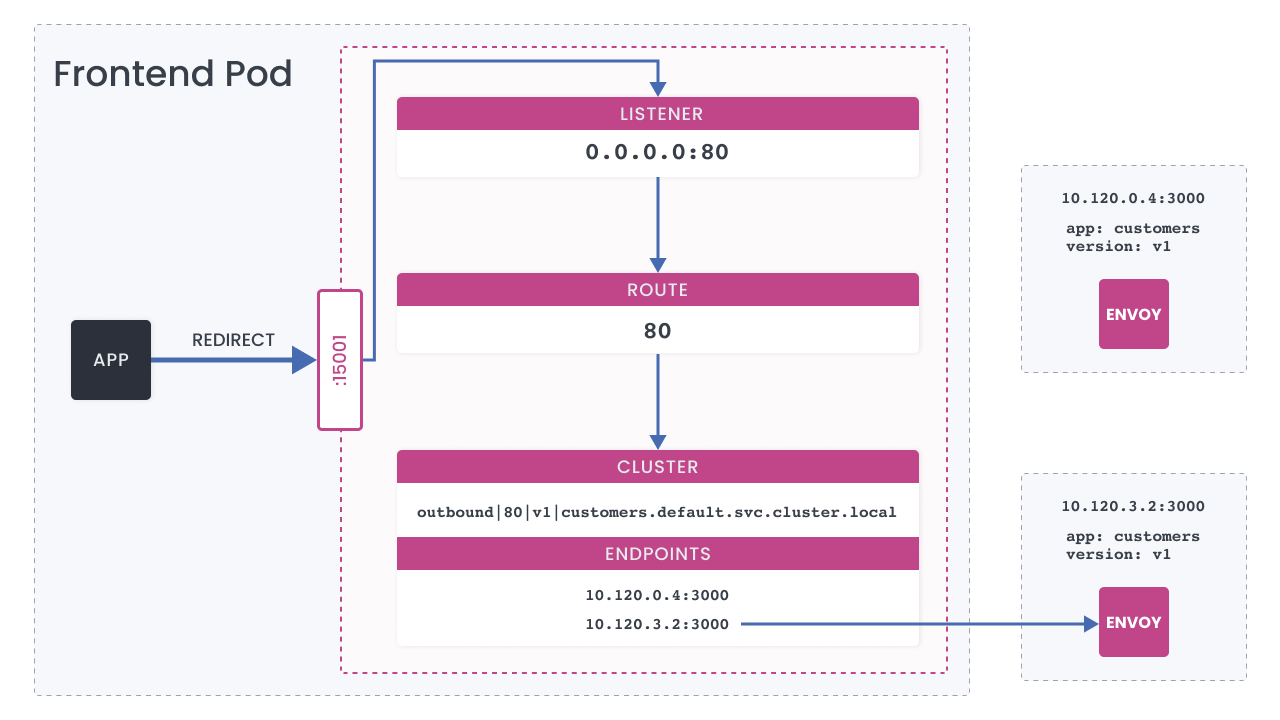
$ istioctl proxy-config endpoints web-frontend-58d497b6f8-lwqkg --cluster "outbound|80|v1|customers.default.svc.cluster.local"

ENDPOINT STATUS OUTLIER CHECK CLUSTER

10.120.0.4:3000 HEALTHY OK outbound|80|v1|customers.default.svc.cluster.local

10.120.3.2:3000 HEALTHY OK outbound|80|v1|customers.default.svc.cluster.local

We can also visualize the above flow using the figure below.

Envoy details

* **listeners.json**12.6 KB

**[DOWNLOADOPENS IN A NEW WINDOW](https://import.cdn.thinkific.com/415508/listeners-210119-085816.json" \t "_blank)**

* **routes.json**11.6 KB

**[DOWNLOADOPENS IN A NEW WINDOW](https://import.cdn.thinkific.com/415508/routes-210119-085816.json" \t "_blank)**

COMPLETE &