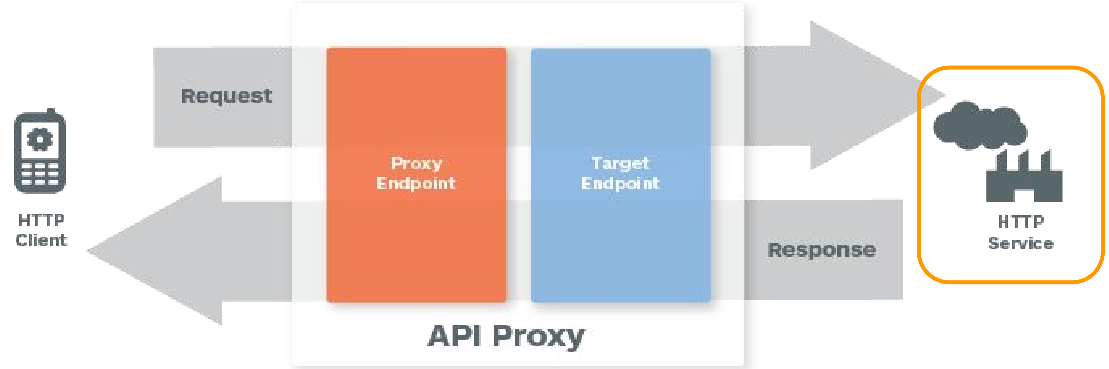
apigee

# Setting Up Named Target Servers

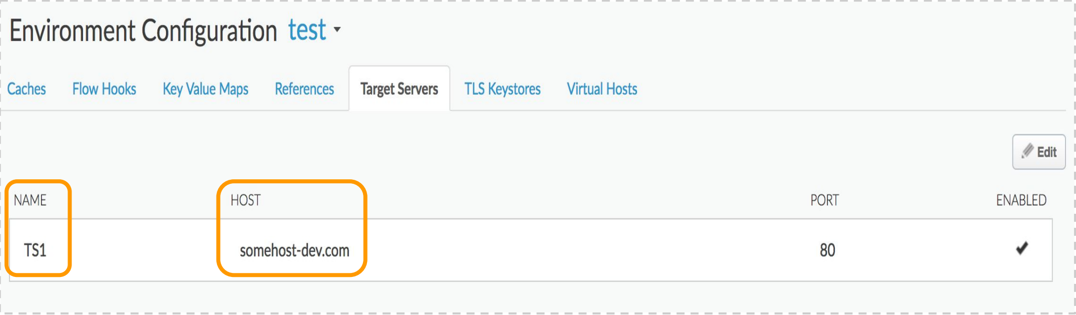
Google Cloud

# Existing Target URL



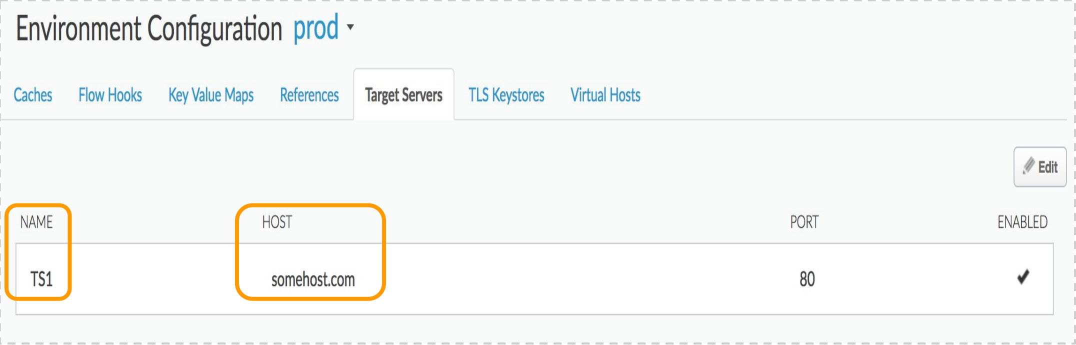
```
<HTTPTargetConnection>
    <URL>http://backend-dev.com</URL>
</HTTPTargetConnection>
```

# Create a Named Target Server



Environment Configuration test ▾

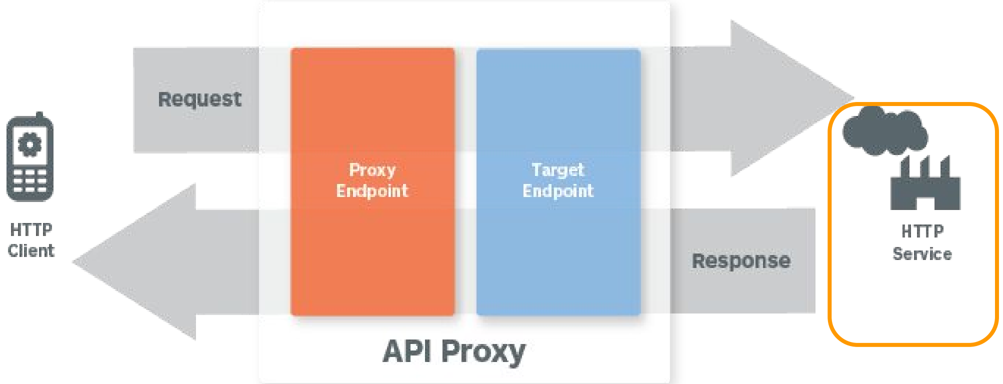
| NAME | HOST | PORT | ENABLED |
|---|---|---|---|
| TS1 | somehost-dev.com | 80 | ✓ |



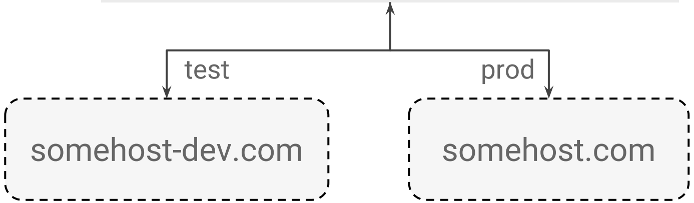
Environment Configuration prod ▾

| NAME | HOST | PORT | ENABLED |
|---|---|---|---|
| TS1 | somehost.com | 80 | ✓ |

# Target Server



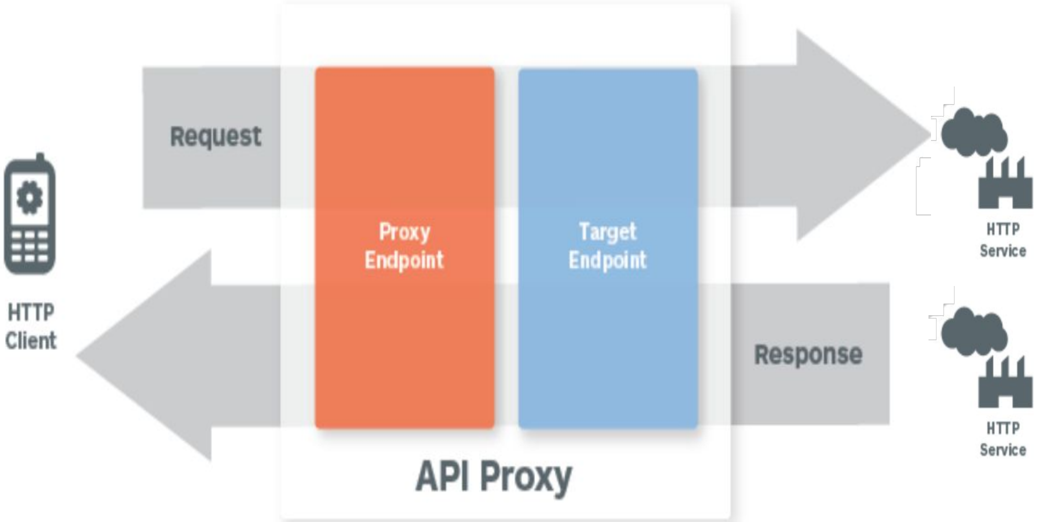
```
<LoadBalancer>
  <Algorithm>RoundRobin</Algorithm>
  <Server name="TS1" />
 </LoadBalancer>
```



# Target Server with SSL

```
<TargetServer name="TS2">
  <IsEnabled>true</IsEnabled>
  <Host>somehost-dev.com</Host>
  <Port>443</Port>
  <SSLInfo>
    <Ciphers/>
    <ClientAuthEnabled>true</ClientAuthEnabled>
    <Enabled>true</Enabled>
    <IgnoreValidationErrors>false</IgnoreValidationErrors>
    <KeyAlias>keystore-alias</KeyAlias>
    <KeyStore>keystore-name</KeyStore>
    <Protocols/>
    <TrustStore>truststore-name</TrustStore>
  </SSLInfo>
</TargetServer>
```

# Multiple Target Servers



# Load Balancer

Algorithms:

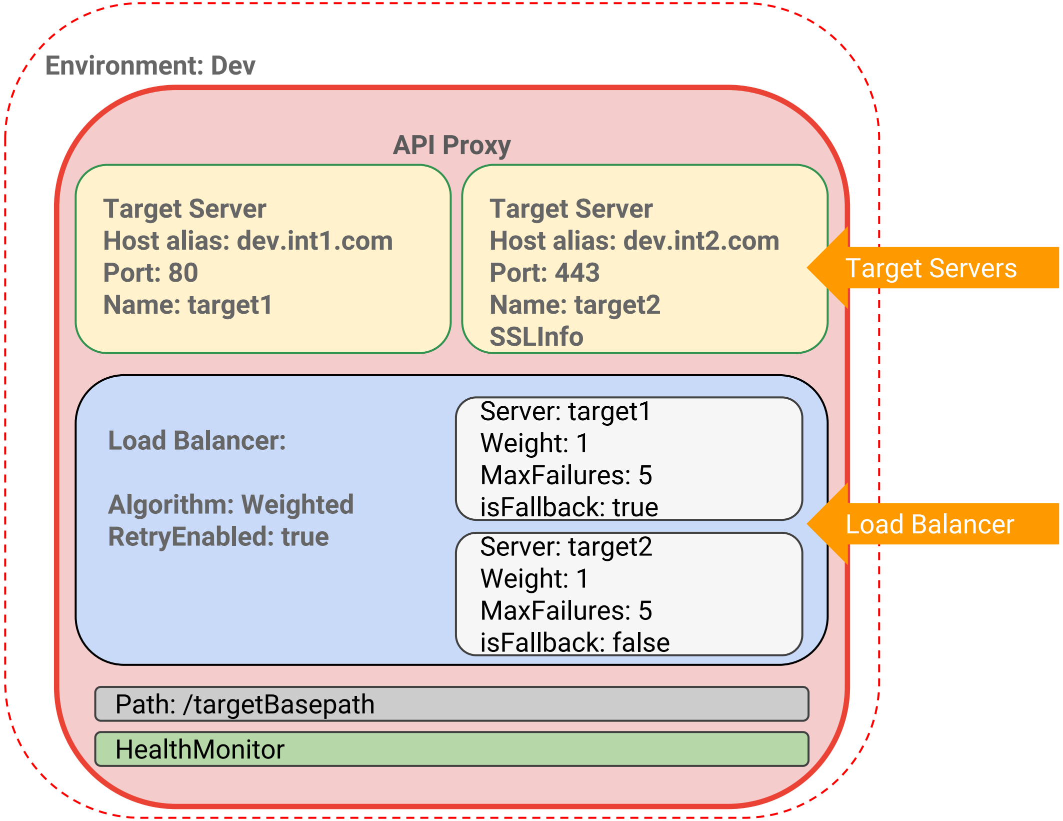
- RoundRobin
- Weighted
- LeastConnections

# Load Balancer

Other Configuration:

- MaxFailures
  - maximum number of failed requests before redirecting to another target server
- RetryEnabled
  - retry after an I/O error
- IsFallback
  - if all others are unavailable, then use fallback server

# Load Balancer



Environment: Dev

API Proxy

Target Server
Host alias: dev.int1.com
Port: 80
Name: target1

Target Server
Host alias: dev.int2.com
Port: 443
Name: target2
SSLInfo

Load Balancer:

Algorithm: Weighted
RetryEnabled: true

Server: target1
Weight: 1
MaxFailures: 5
isFallback: true

Server: target2
Weight: 1
MaxFailures: 5
isFallback: false

Path: /targetBasepath

HealthMonitor

# Load Balancer

## Round Robin / Least Connections

```
<TargetEndpoint name="default">
  <HTTPTargetConnection>
    <LoadBalancer>
    <Algorithm>RoundRobin</Algorithm>
      <Server name="target1" />
      <Server name="target2" />
    </LoadBalancer>
    <Path>/test</Path>
  </HTTPTargetConnection>
</TargetEndpoint>
```

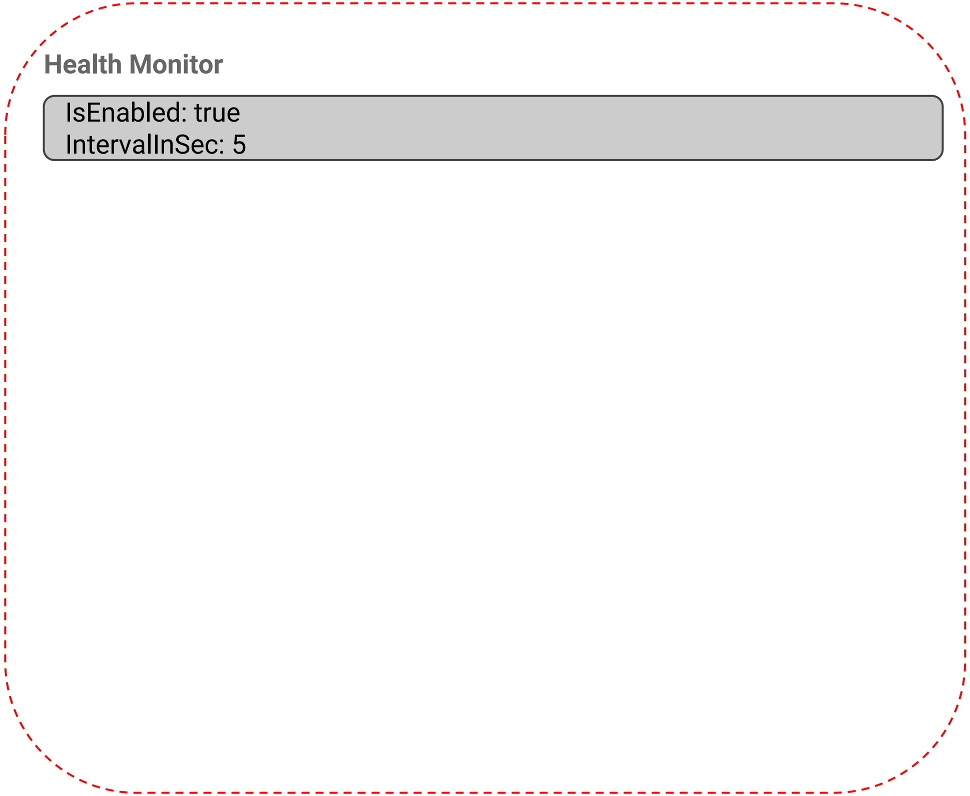
```
<TargetEndpoint name="default">
  <HTTPTargetConnection>
    <LoadBalancer>
    <Algorithm>LeastConnections</Algorithm>
      <Server name="target1" />
      <Server name="target2" />
    </LoadBalancer>
    <Path>/test</Path>
  </HTTPTargetConnection>
</TargetEndpoint>
```

# Load Balancer

## Weighted

```
<TargetEndpoint name="default">
  <HTTPTargetConnection>
    <LoadBalancer>
      <Algorithm>Weighted</Algorithm>
      <Server name="target1">
        <Weight>1</Weight>
      </Server>
      <Server name="target2">
        <Weight>2</Weight>
      </Server>
    </LoadBalancer>
    <Path>/test</Path>
  </HTTPTargetConnection>
</TargetEndpoint>
```

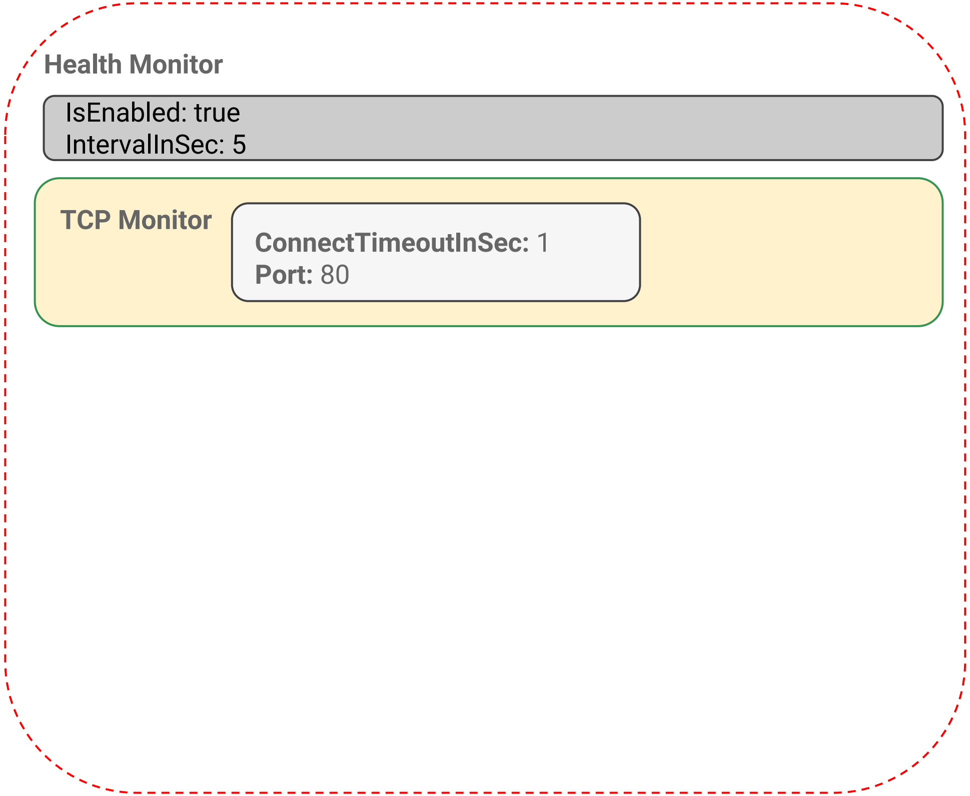
# Health Monitor



Health Monitor

IsEnabled: true
IntervalInSec: 5

# Health Monitor



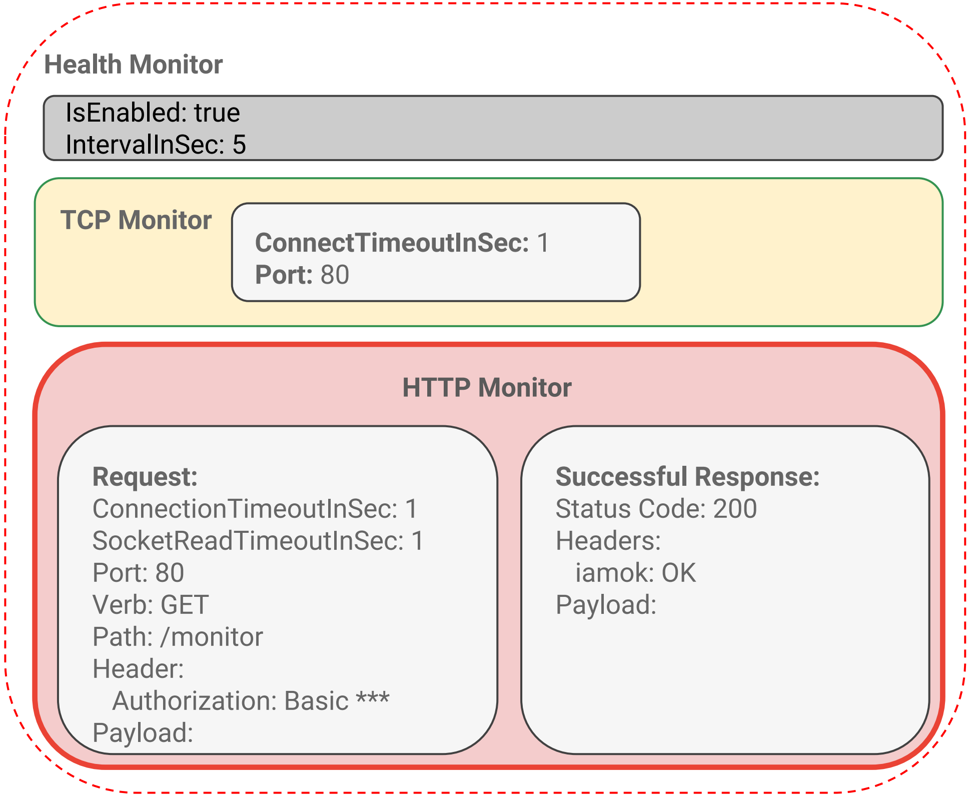
**Health Monitor**

IsEnabled: true
IntervalInSec: 5

**TCP Monitor**

**ConnectTimeoutInSec:** 1
**Port:** 80

# Health Monitor



**Health Monitor**

IsEnabled: true
IntervalInSec: 5

**TCP Monitor**

**ConnectTimeoutInSec:** 1
**Port:** 80

**HTTP Monitor**

**Request:**
ConnectionTimeoutInSec: 1
SocketReadTimeoutInSec: 1
Port: 80
Verb: GET
Path: /monitor
Header:
  Authorization: Basic ***
Payload:

**Successful Response:**
Status Code: 200
Headers:
  iamok: OK
Payload:

# Health Monitor - HTTPMonitor

```
<HealthMonitor>
  <IsEnabled>true</IsEnabled>
  <IntervalInSec>5</IntervalInSec>
  <HTTPMonitor>
    <Request>
      <ConnectTimeoutInSec>10</ConnectTimeoutInSec>
      <SocketReadTimeoutInSec>30</SocketReadTimeoutInSec>
      <Port>80</Port>
      <Verb>GET</Verb>
      <Path>/healthcheck</Path>
      <Header name="Authorization">Basic ****</Header>
    </Request>
    <SuccessResponse>
      <ResponseCode>200</ResponseCode>
      <Header name="ImOK">YourOK</Header>
    </SuccessResponse>
  </HTTPMonitor>
</HealthMonitor>
```

# Health Monitor - TCPMonitor

```
<HealthMonitor>
  <IsEnabled>true</IsEnabled>
  <IntervalInSec>5</IntervalInSec>
  <TCPMonitor>
   <ConnectTimeoutInSec>10</ConnectTimeoutInSec>
   <Port>80</Port>
  </TCPMonitor>
</HealthMonitor>
```

apigee

# Thank You

Google Cloud