Implementing Load Balancing with Instance Groups

Vitthal Srinivasan
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Introducing instance templates

Managed instance groups and unmanaged instance groups

Architectural overview of the HTTP(S) load balancer and its components

Cross-regional load balancers with unmanaged and managed instance groups

Autoscaling with managed instance groups

Instance Groups

Instance Groups

A group of machines which can be created and managed together to avoid individually controlling each instance in the group

Two Kinds of Instance Groups

Managed

Unmanaged

Two Kinds of Instance Groups

Managed

Unmanaged

Unmanaged

Groups of dissimilar instances that you can add and remove from the group

Do not offer autoscaling, rolling updates or instance templates

Not recommended, used only when you need to apply load balancing to pre-existing configurations

Two Kinds of Instance Groups

Managed

Unmanaged

Managed

Uses an instance template to create a group of identical instances

Changes to the instance group changes all instances in the group

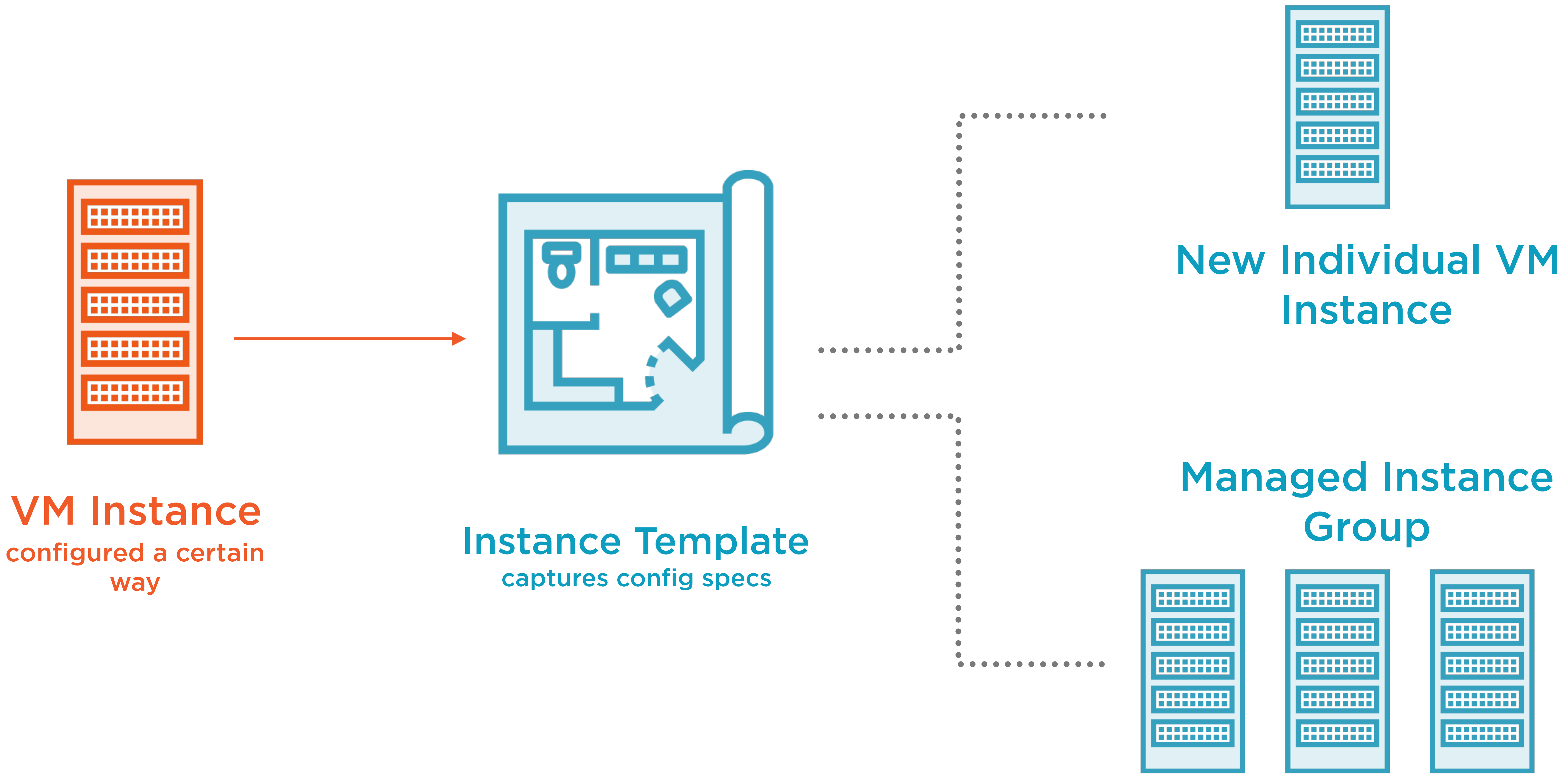
Managed Instance Group

Group of identical GCE VM instances, created from the same instance template that are managed by the platform

Instance Template

A specification of machine type, boot disk (or container image), zone, labels and other instance properties that can be used to instantiate either individual VM instances or a Managed Instance Group

Instance Template



Attractions of MIGs

Autoscaling

Associate autoscaling policy with MIG

Autohealing

Associate health check and autohealing policy with MIG

Autoscaling and Autohealing

Attractions of MIGs

Autoscaling

Associate autoscaling policy with MIG

Autohealing

Associate health check and autohealing policy with MIG

Attractions of MIGs

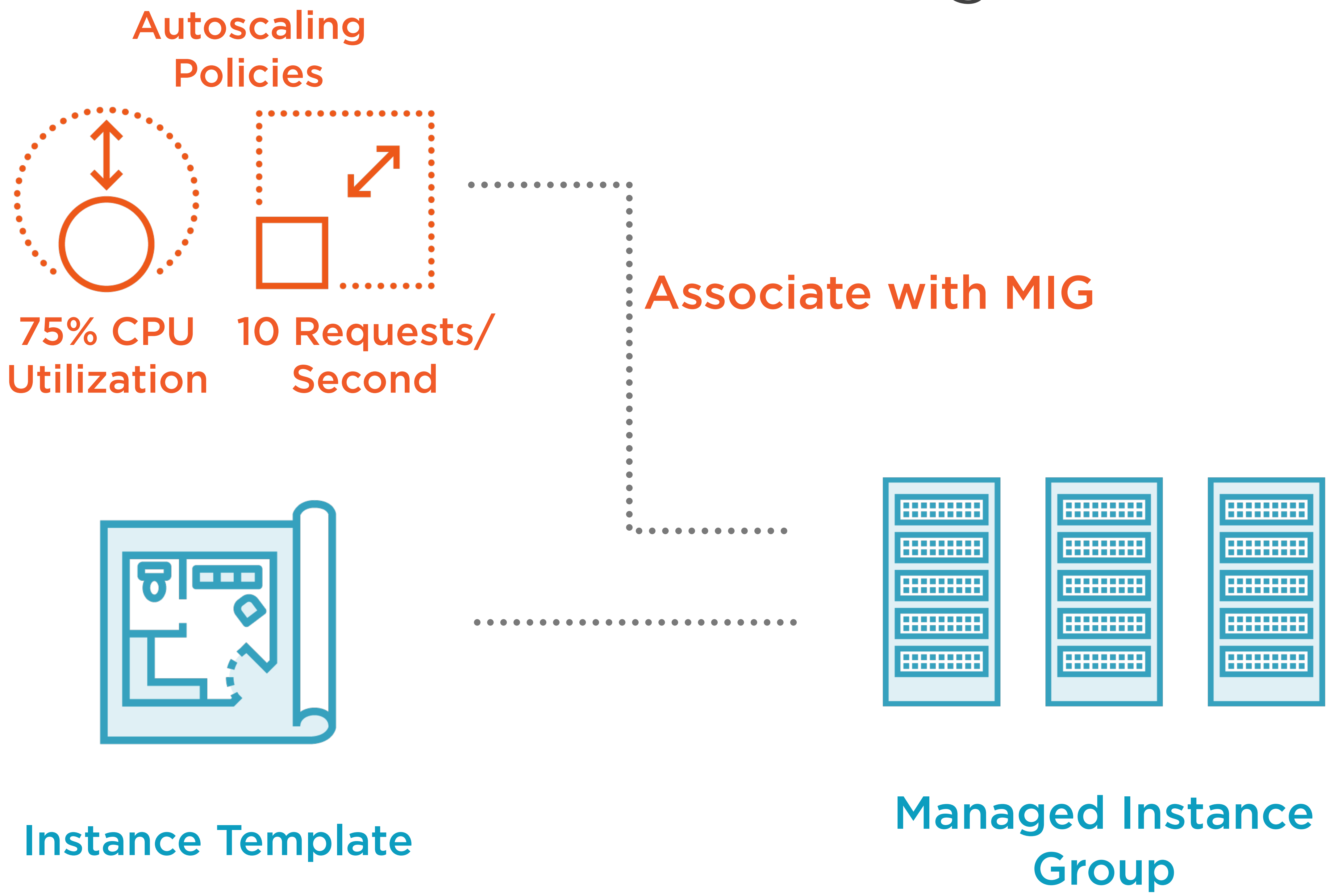
Autoscaling

Associate autoscaling policy with MIG

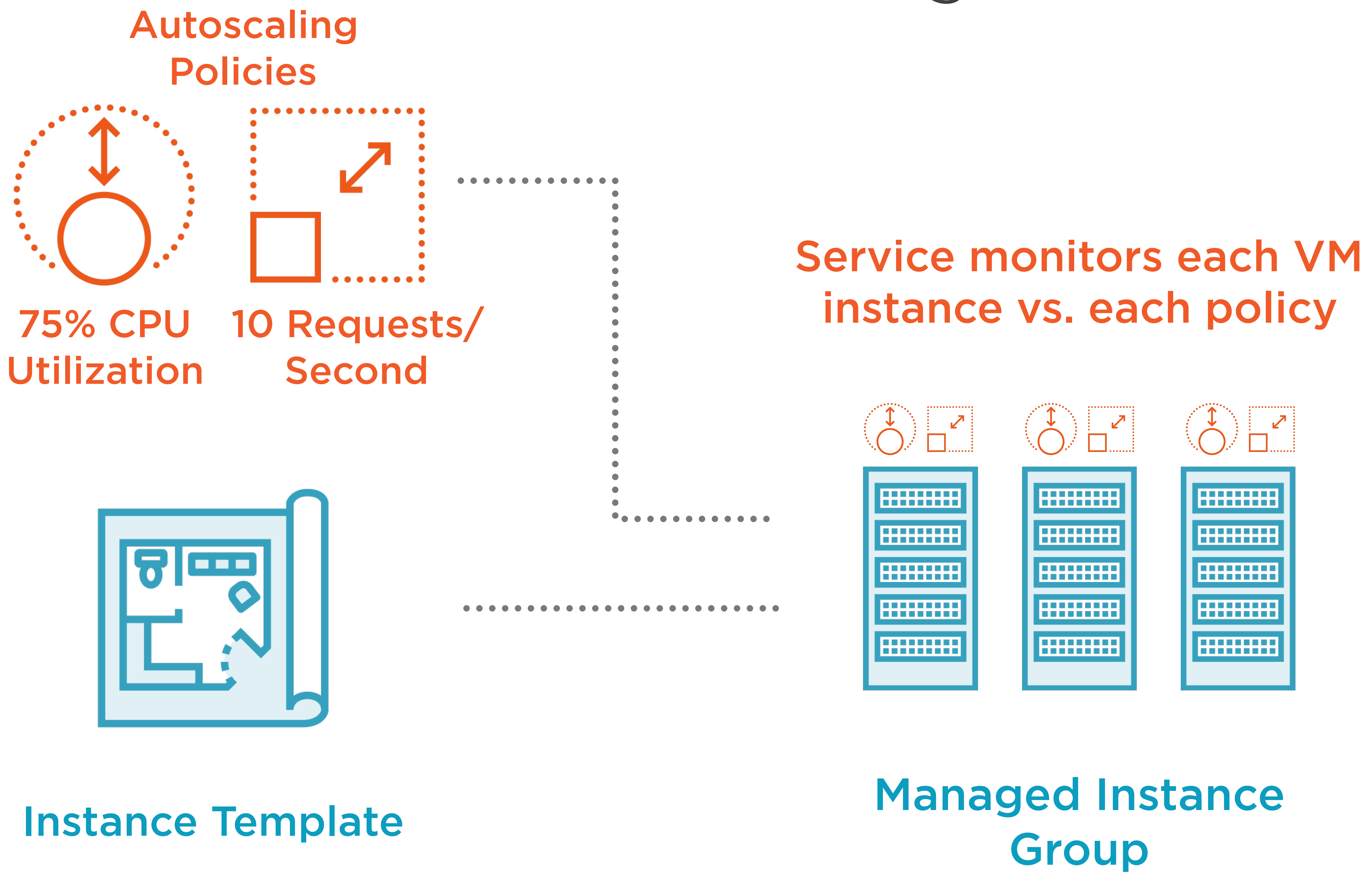
Autohealing

Associate health check and autohealing policy with MIG

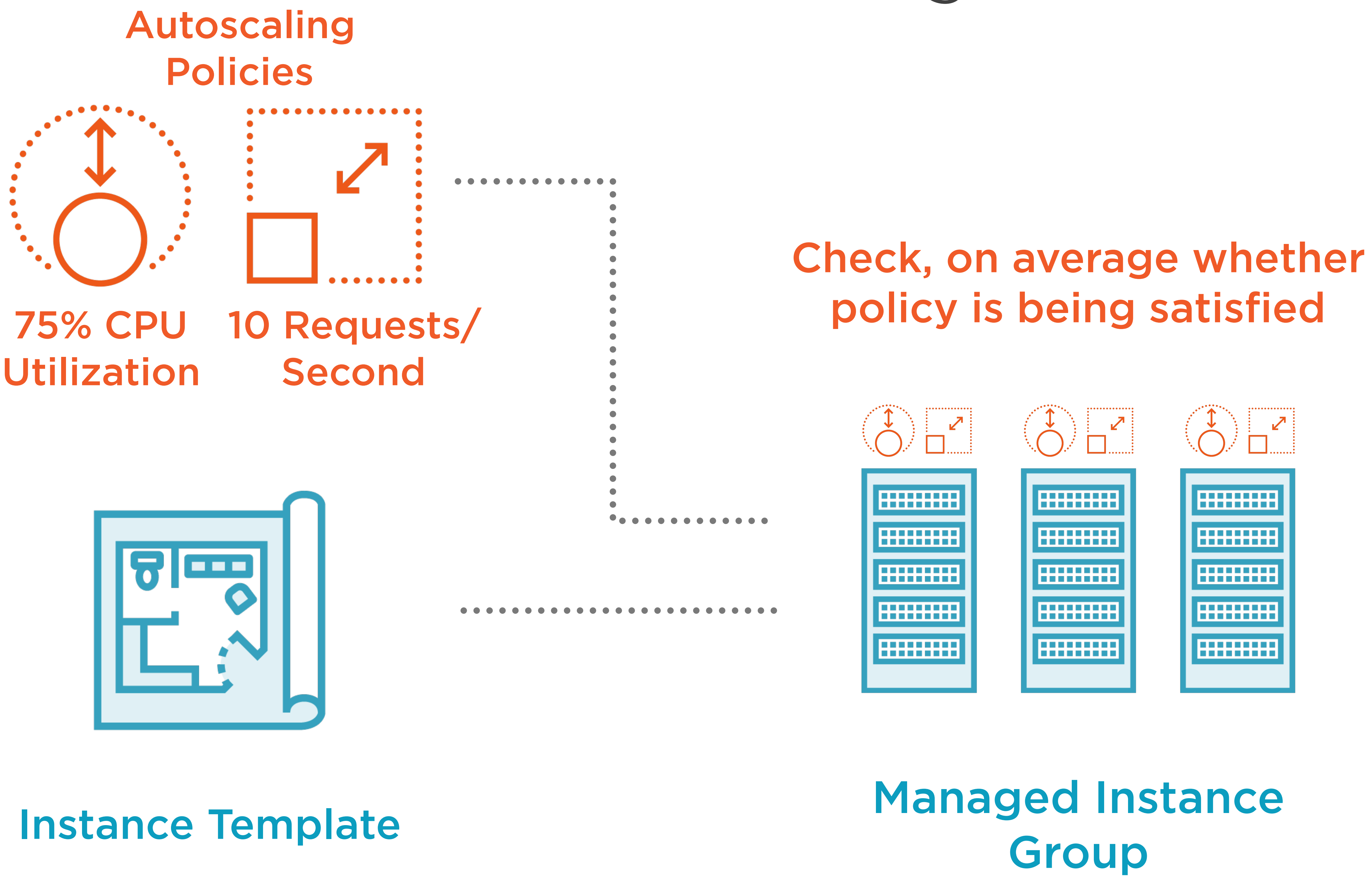
Autoscaling Policies



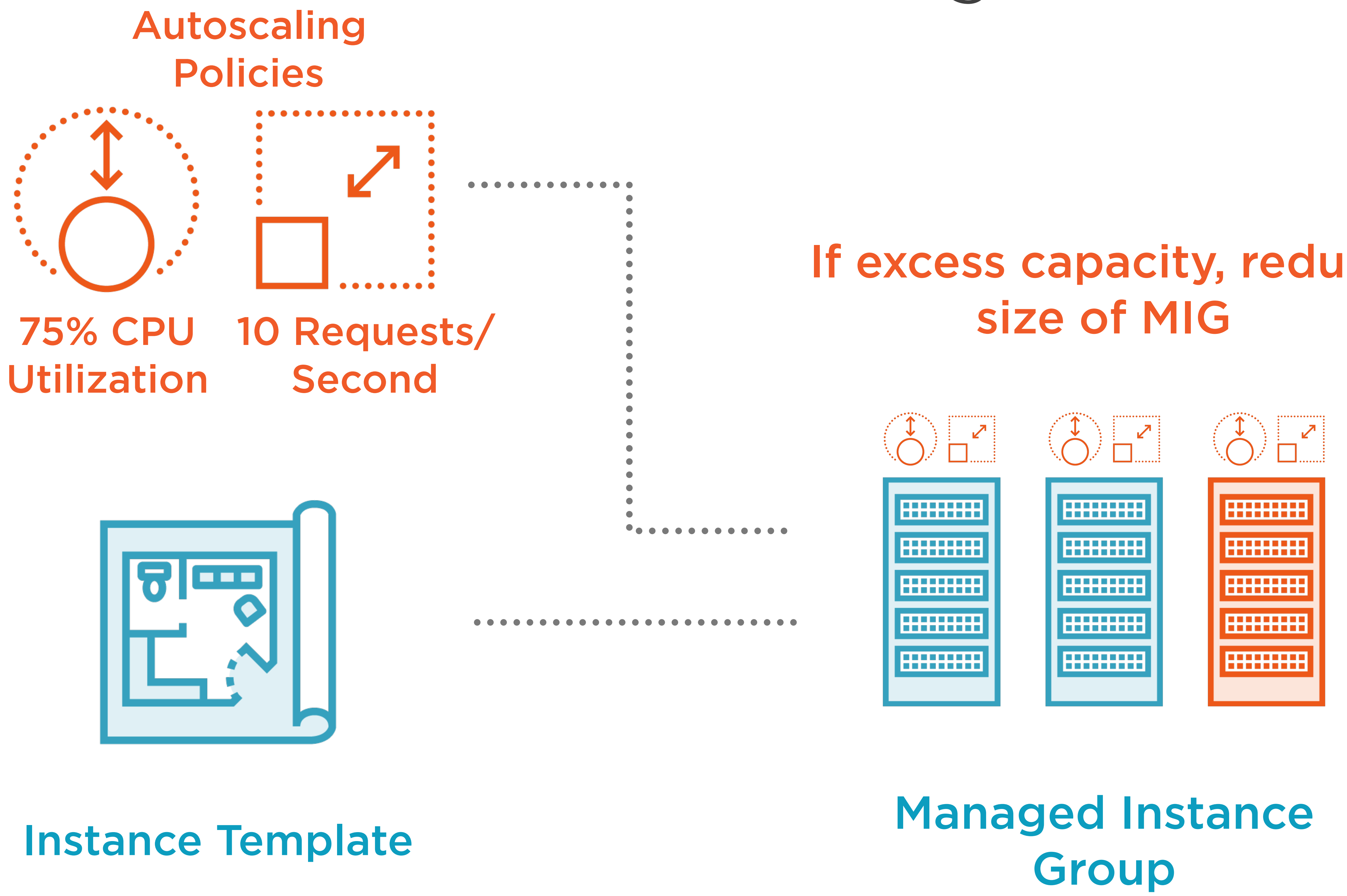
Autoscaling Policies



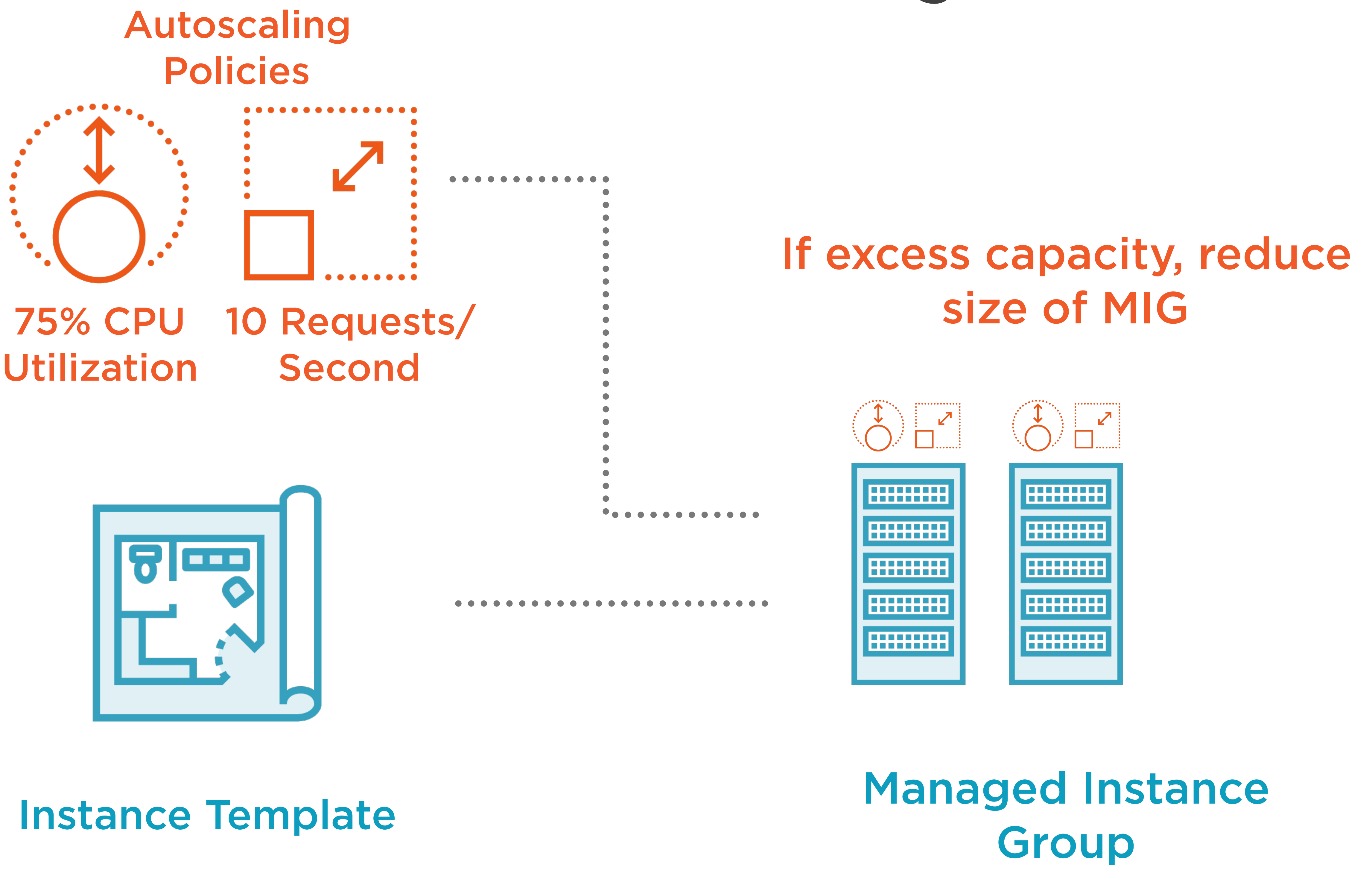
Autoscaling Policies



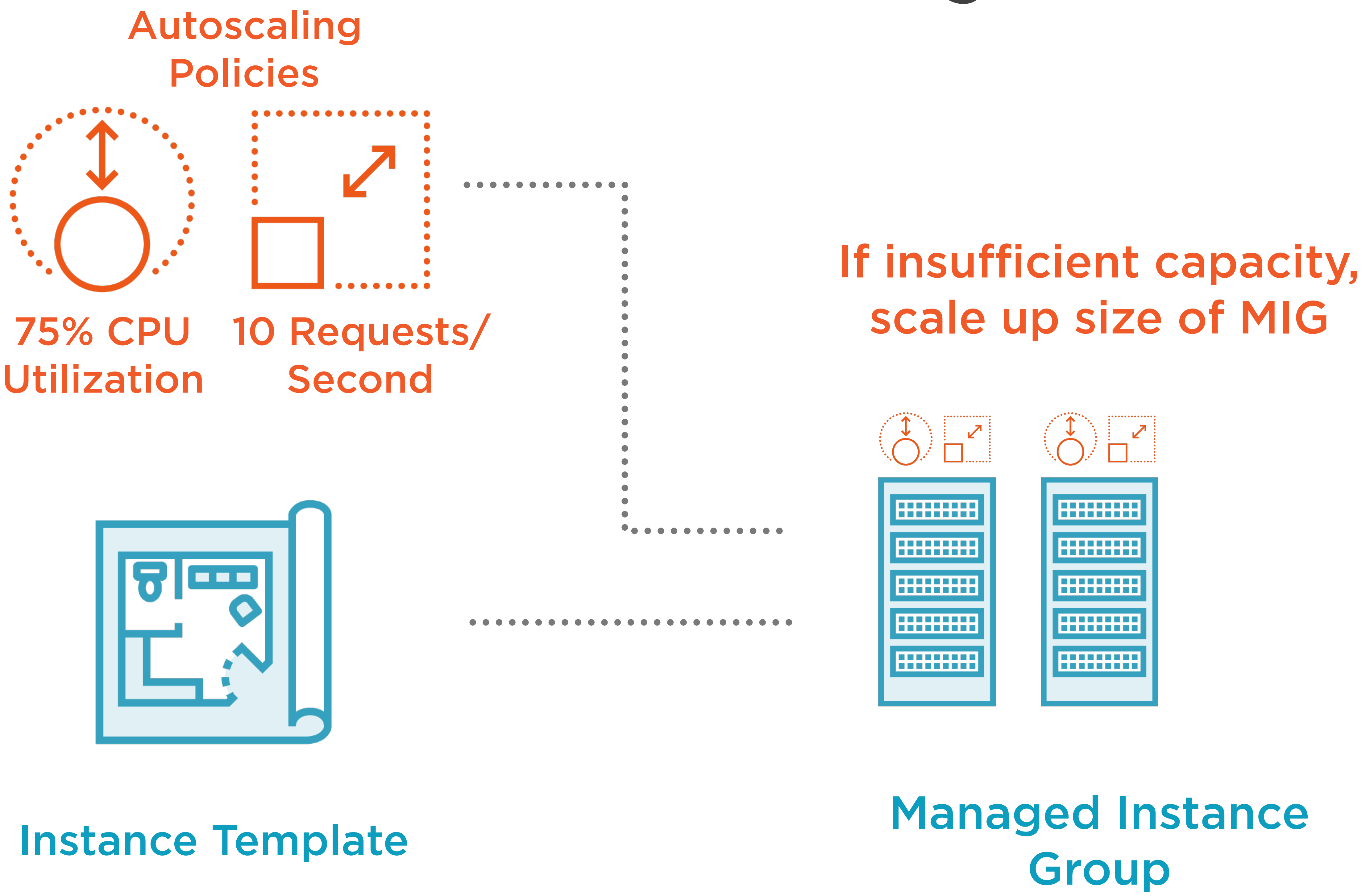
Autoscaling Policies



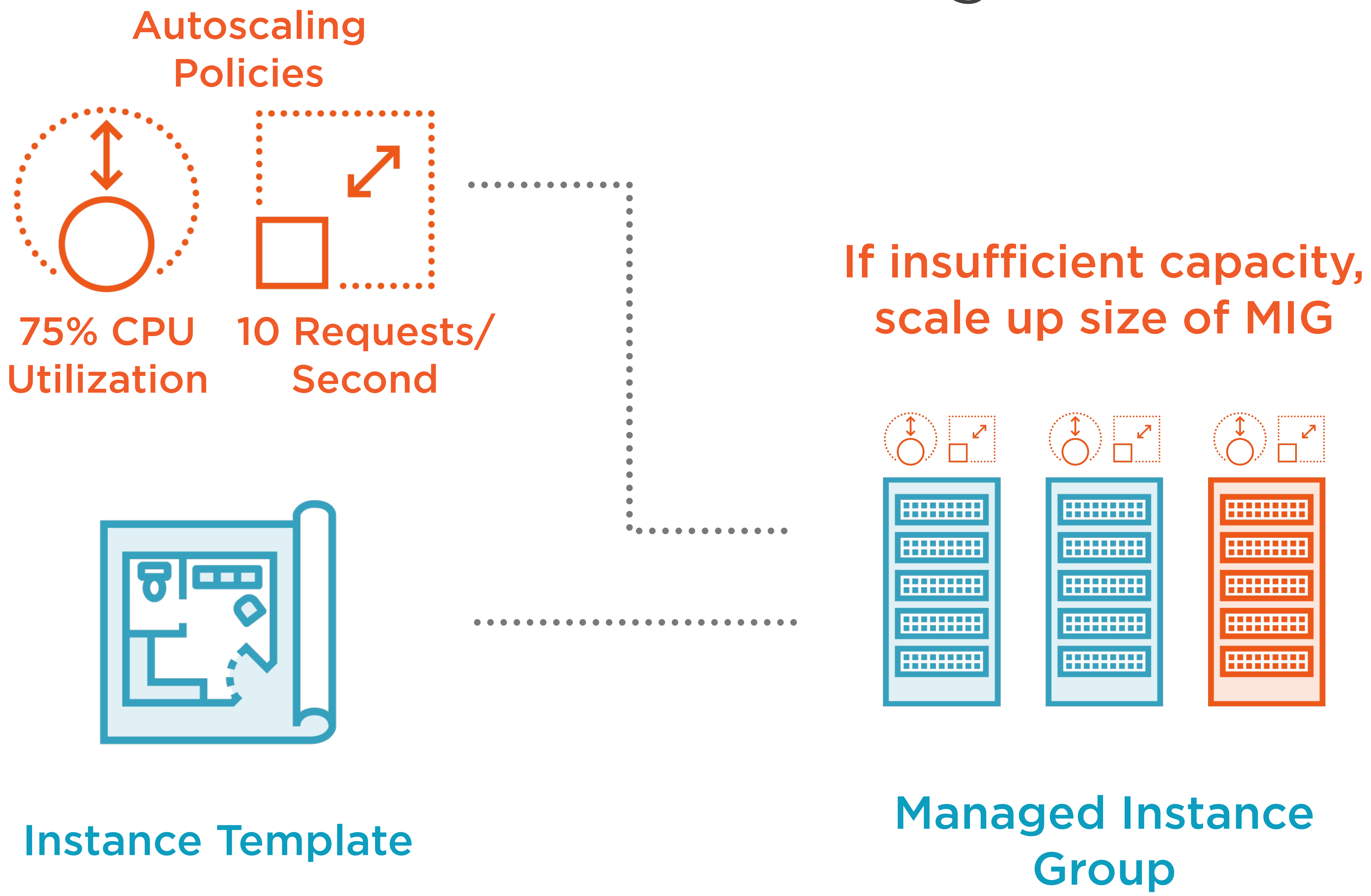
Autoscaling Policies



Autoscaling Policies



Autoscaling Policies



Attractions of MIGs

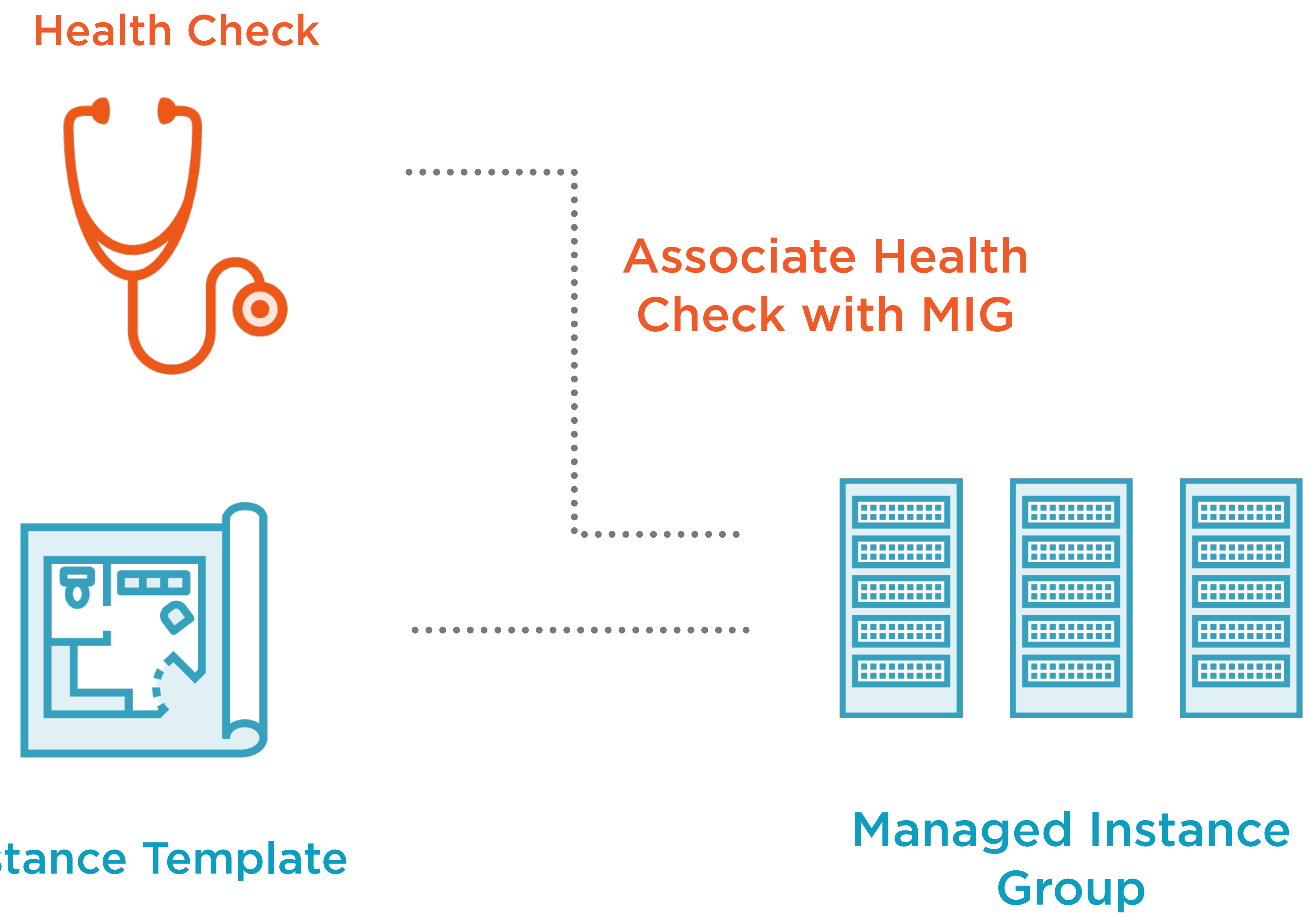
Autoscaling

Associate autoscaling policy with MIG

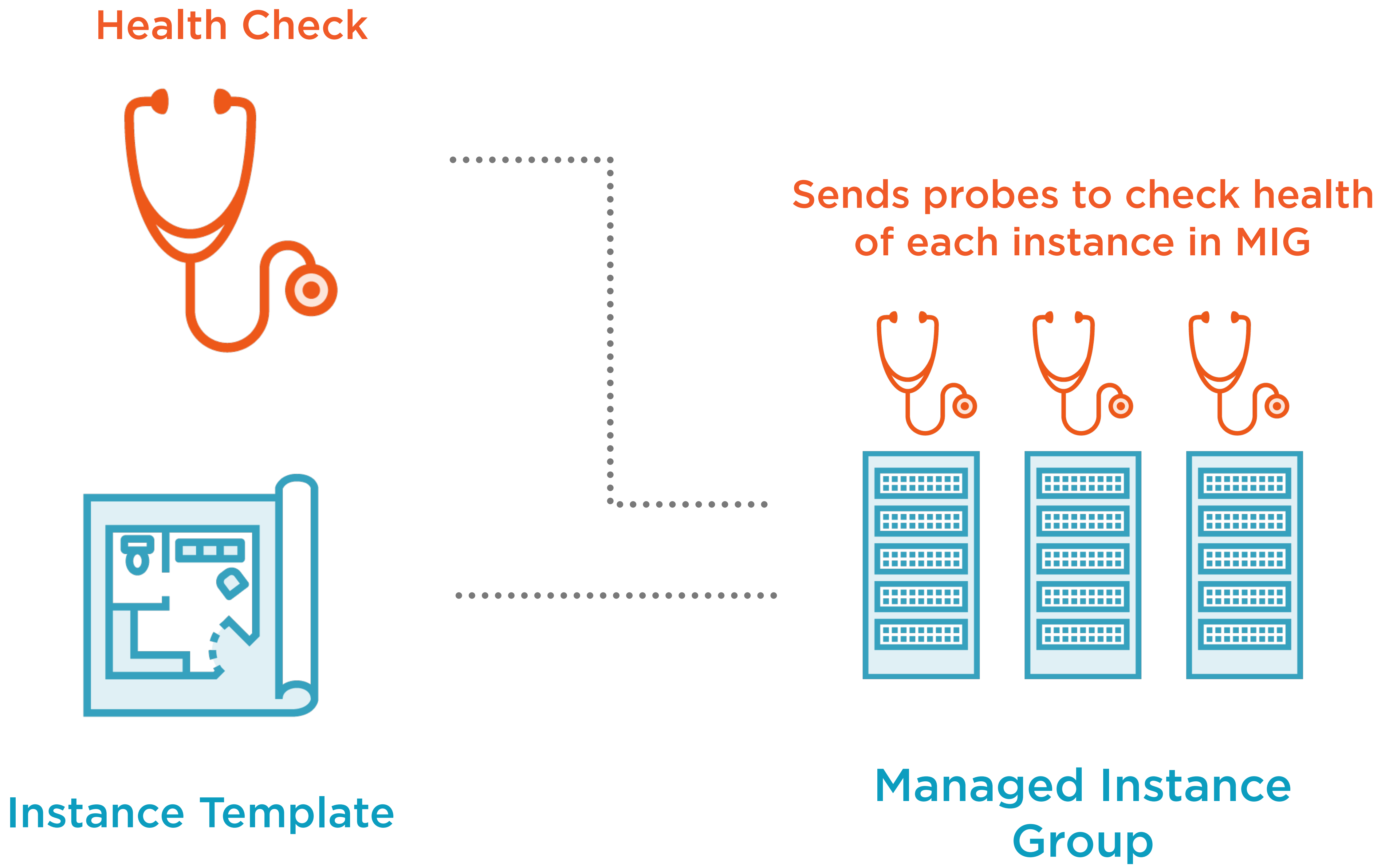
Autohealing

Associate health check and autohealing policy with MIG

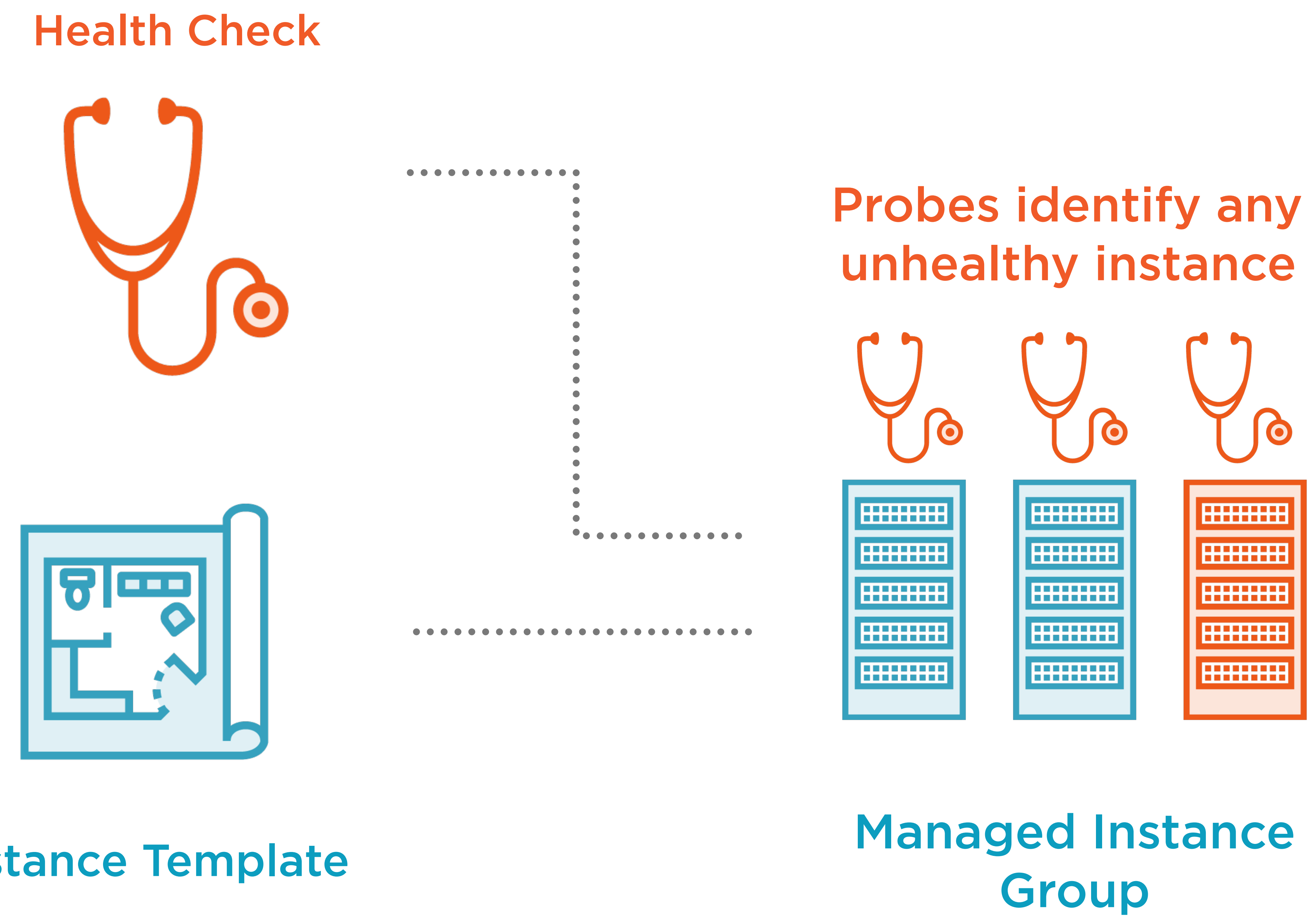
Health Checks



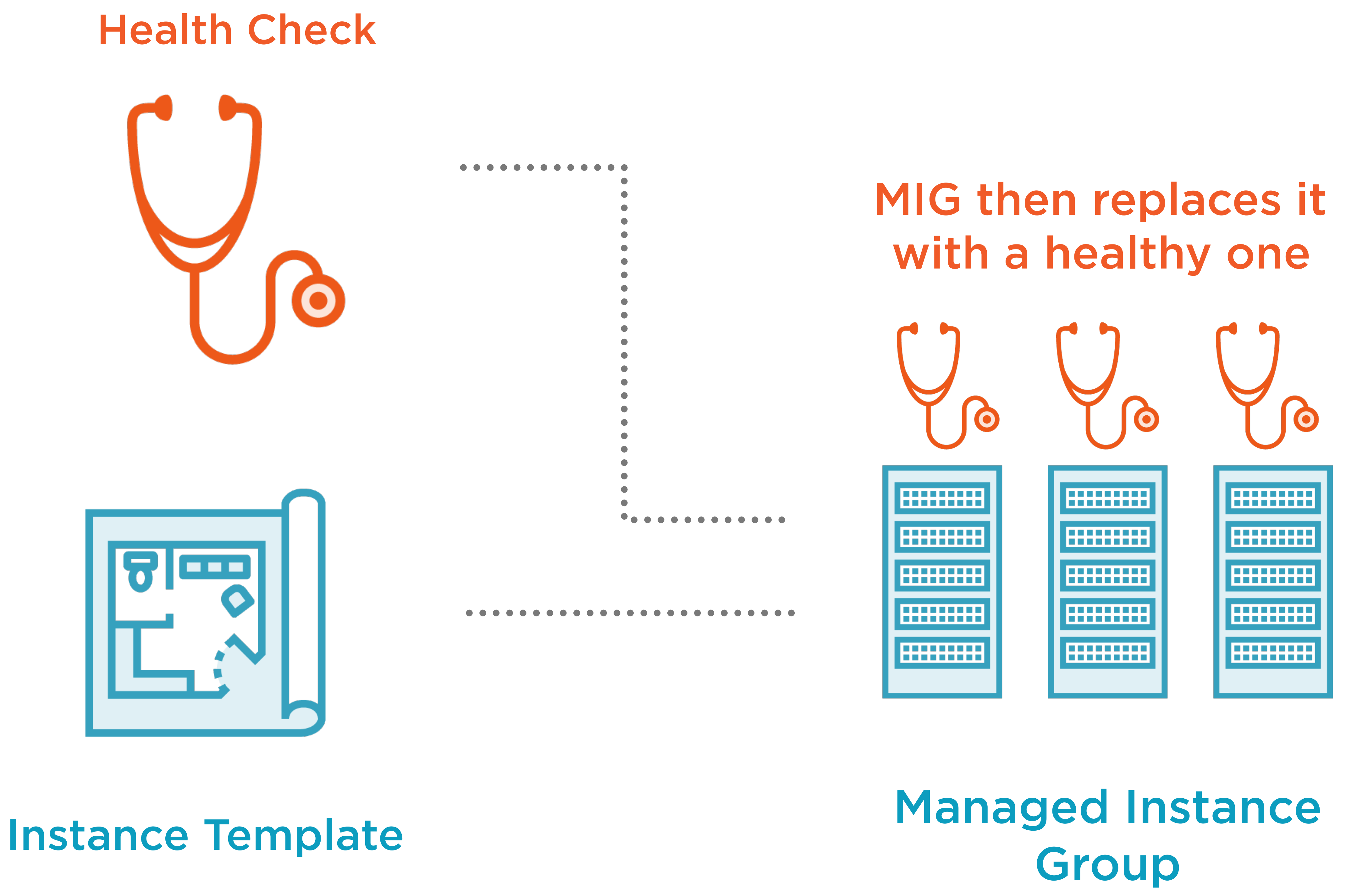
Health Checks



Health Checks

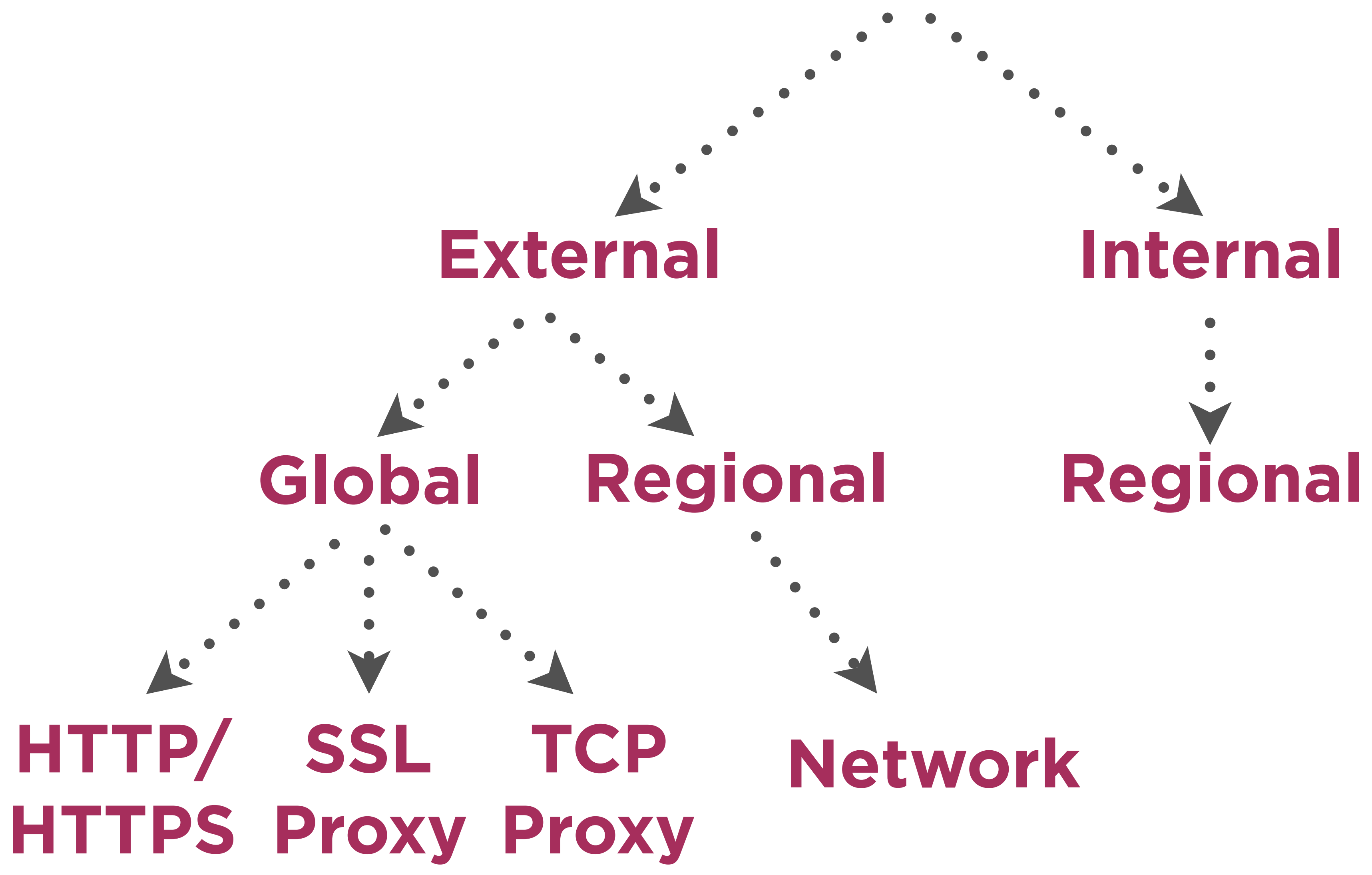


Health Checks

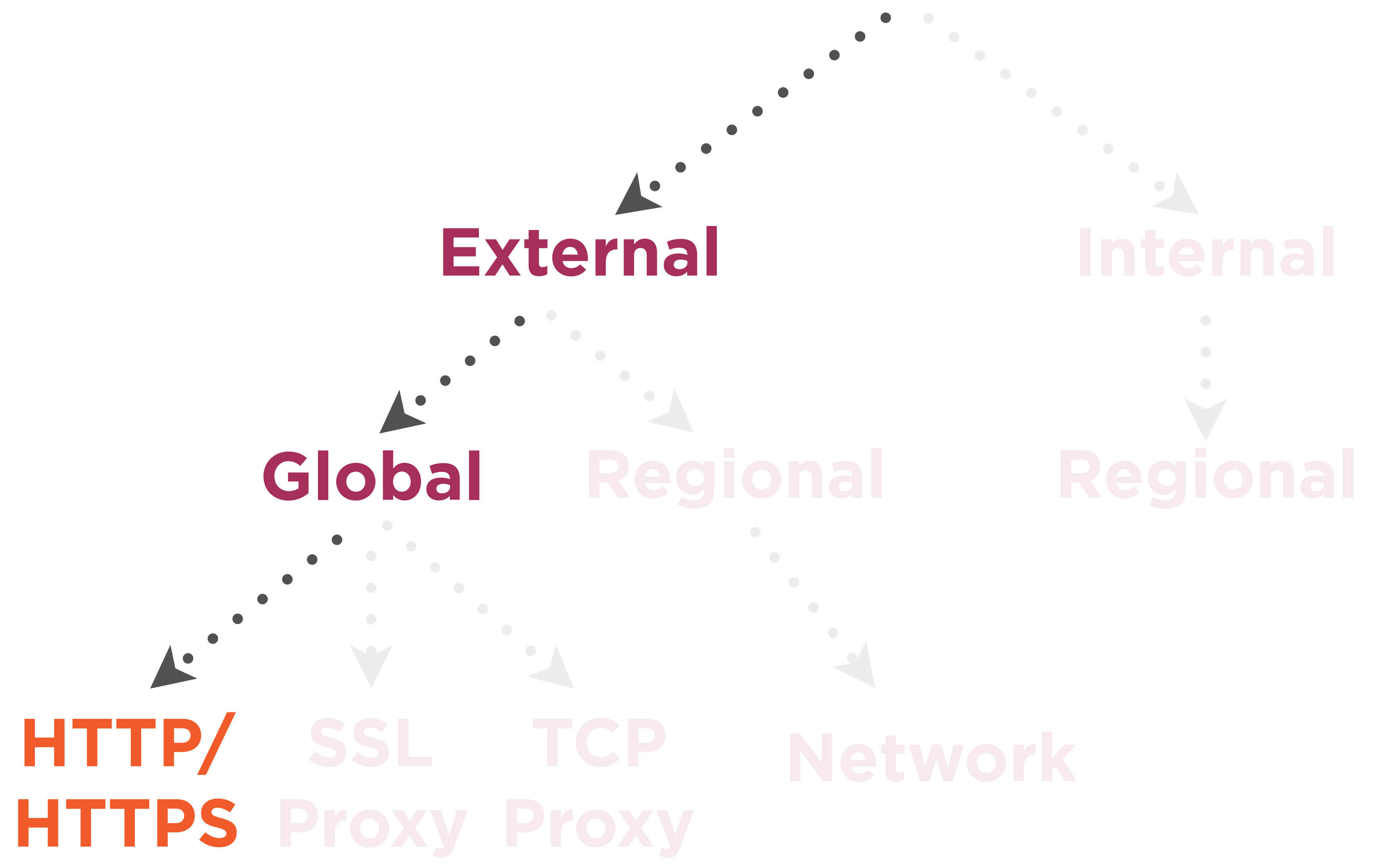


HTTP(S) Load Balancing

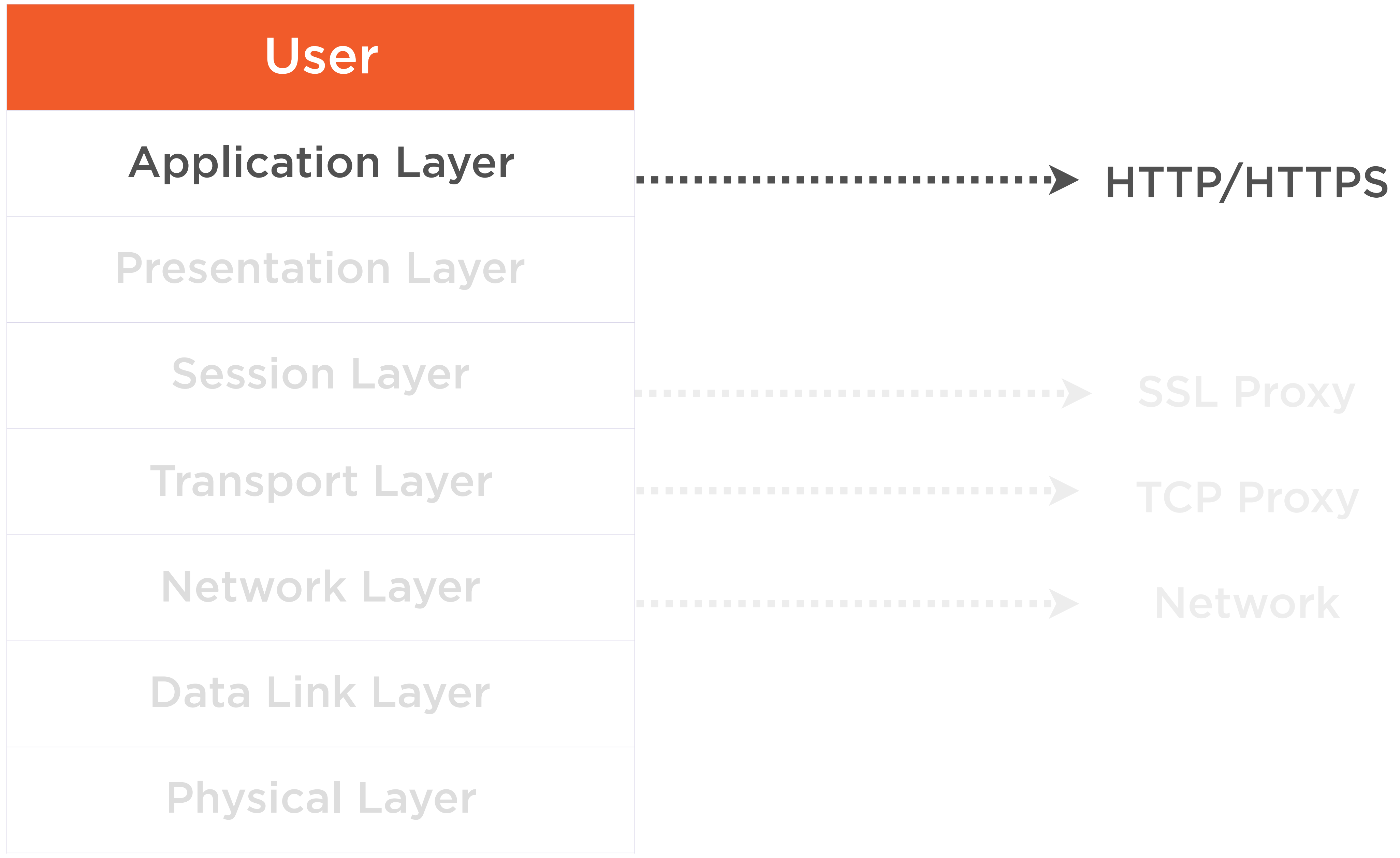
Load Balancing



HTTP(S) Load Balancing



HTTP(S) Load Balancing



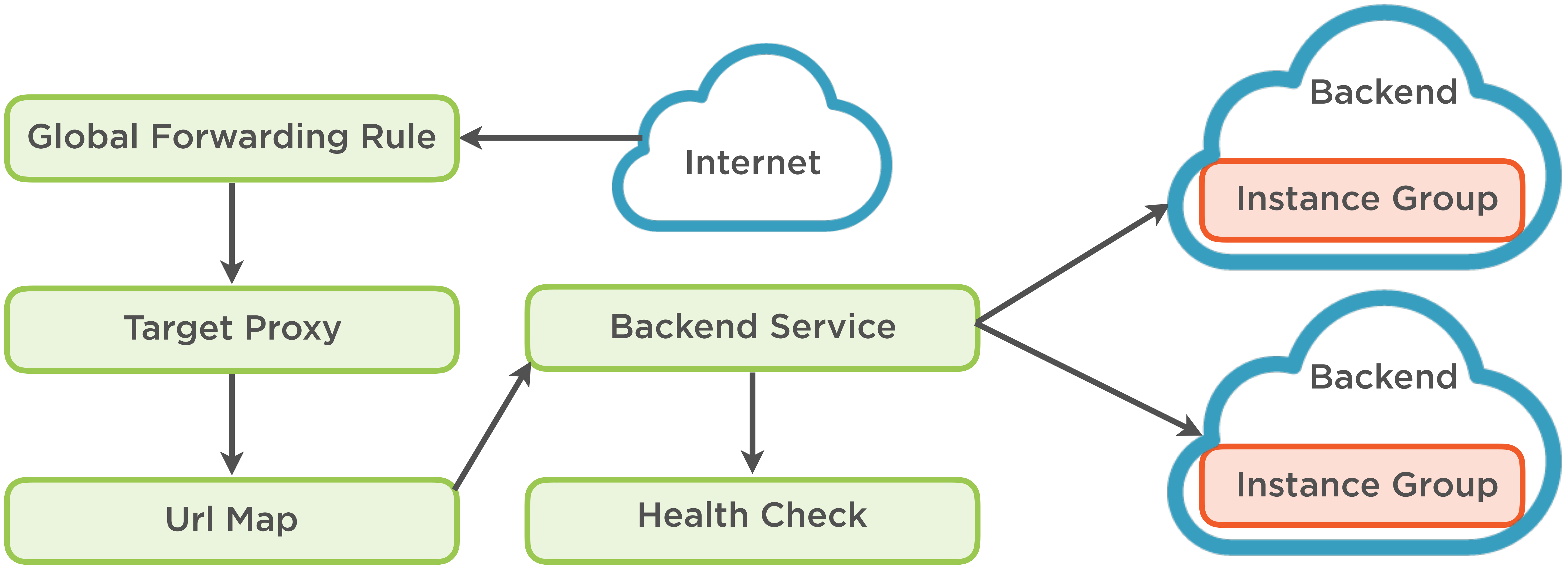
HTTP(S) is used to balance global, external traffic

HTTP(S) Load Balancing

Distributes HTTP(S) traffic among groups of instances based on:

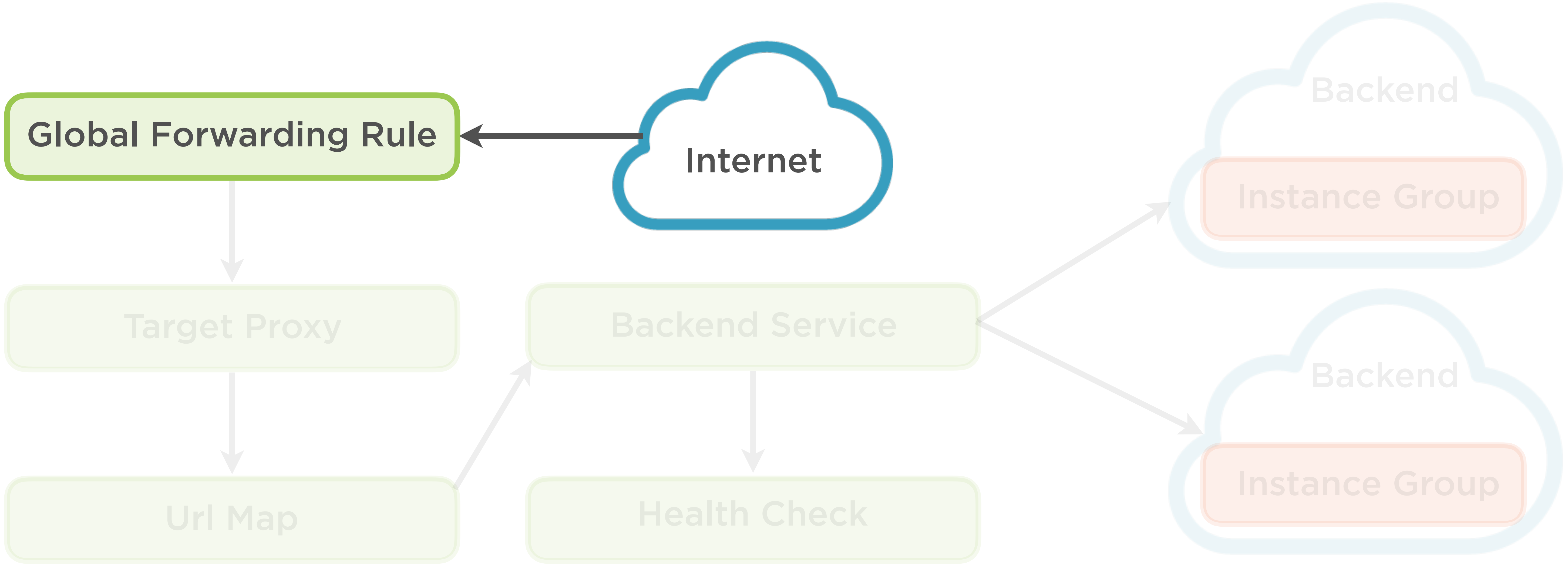
- Proximity to the user
- Requested URL
- Or both.

HTTP(S) Load Balancing



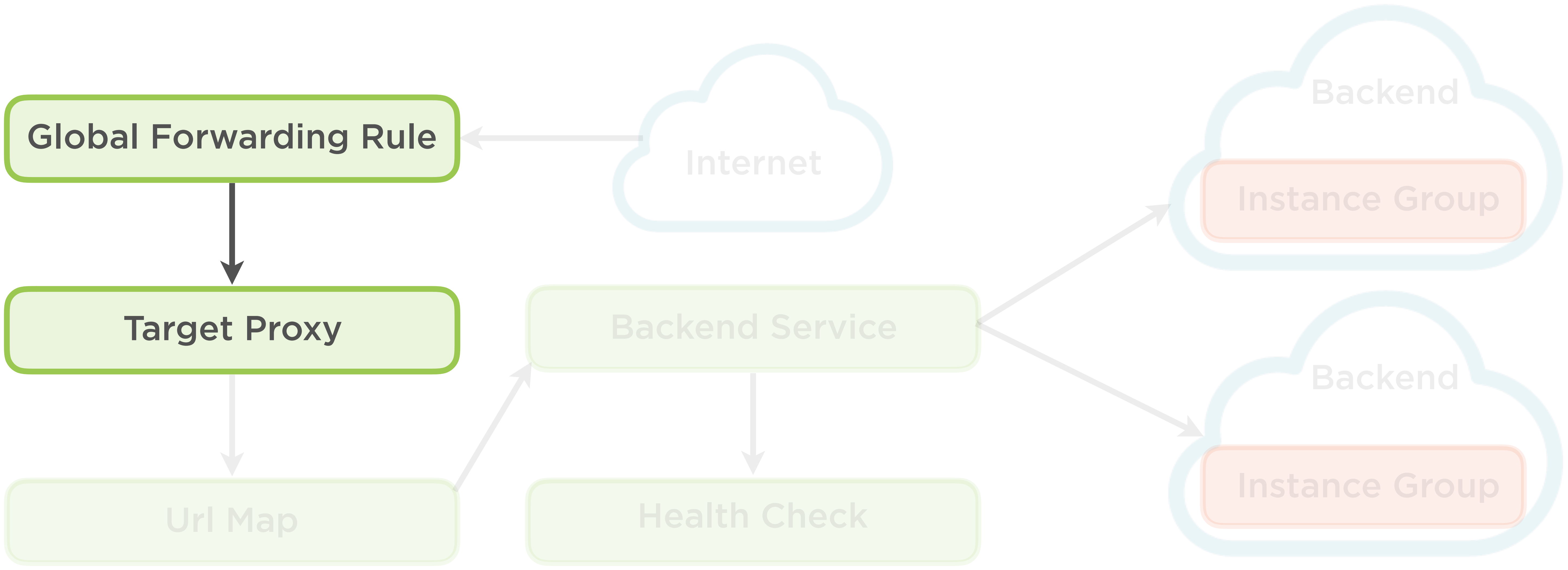
A global, external load balancing service offered on the GCP

Global Forwarding Rule



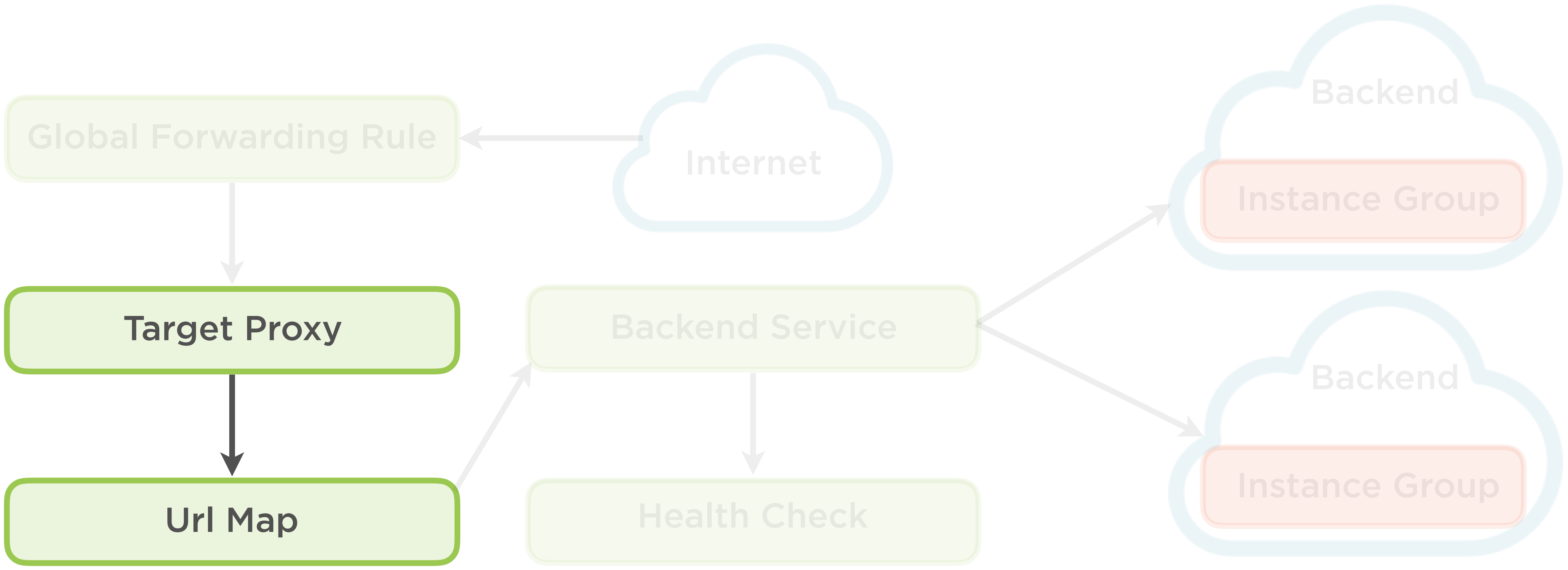
Traffic from the internet is sent to a global forwarding rule - this rule determines which proxy the traffic should be directed to

Target Proxy



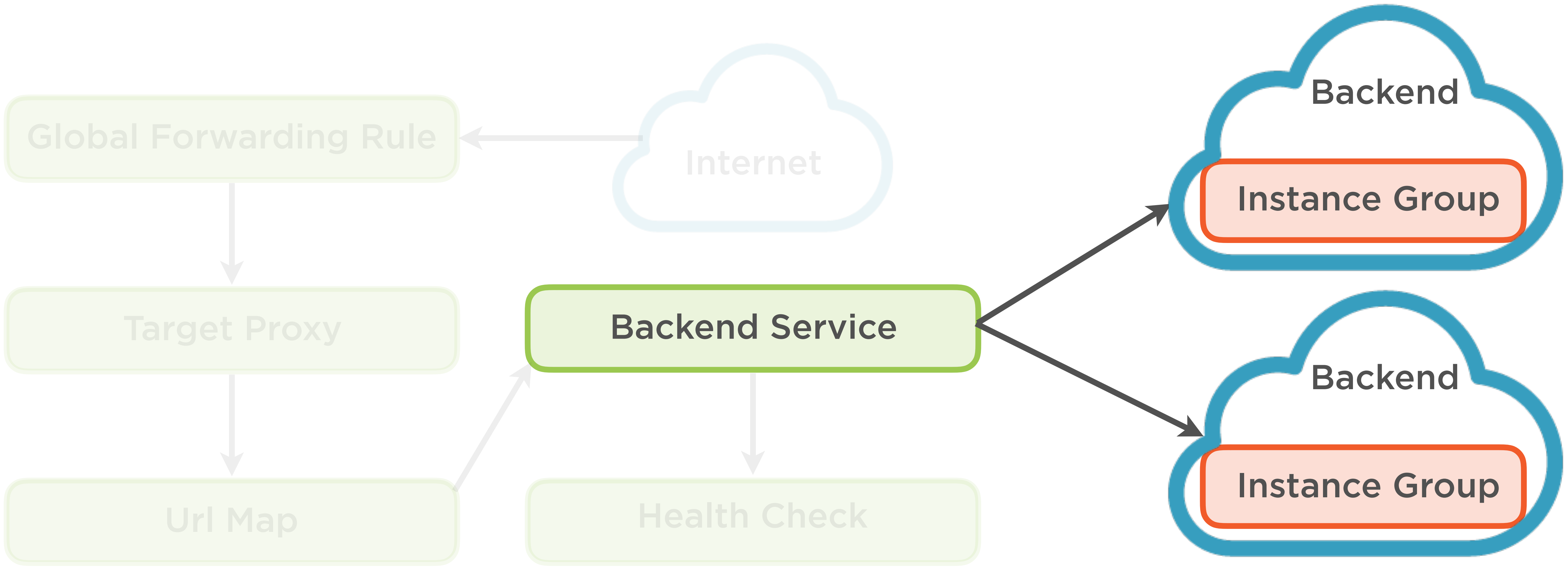
The global forwarding rule directs incoming requests to a target HTTP proxy

URL Map



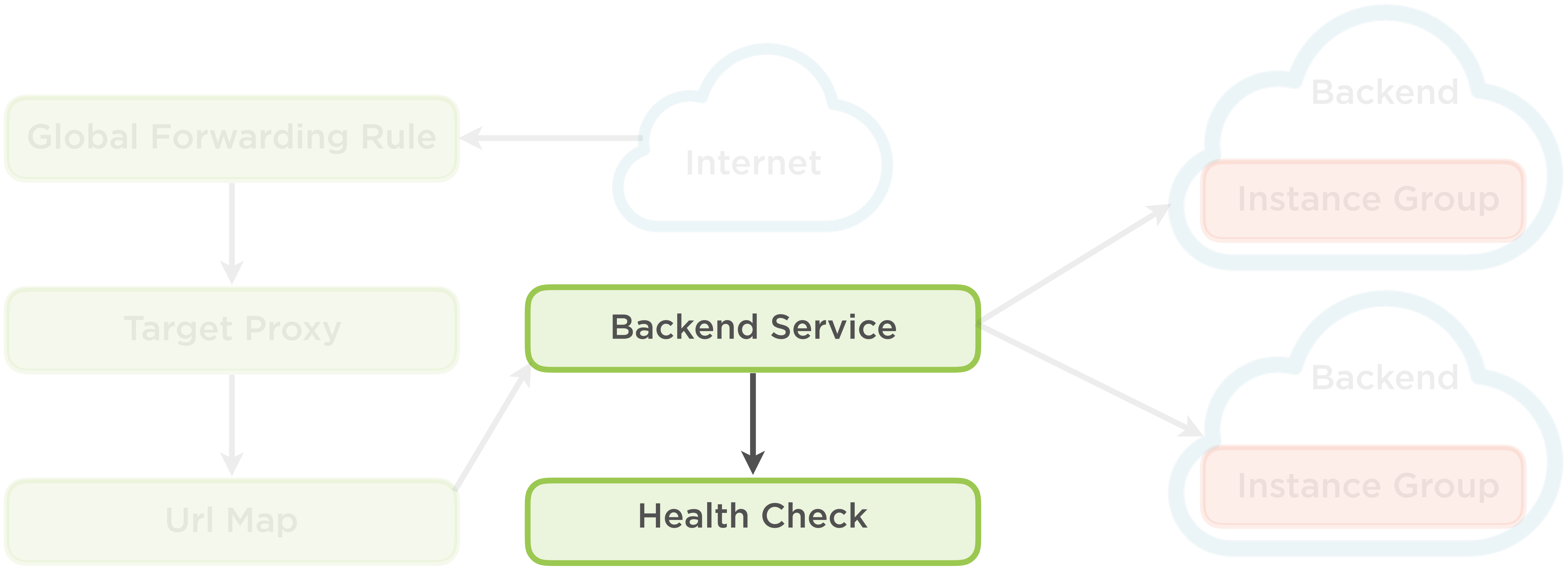
The target HTTP proxy checks each request against a URL map to determine the appropriate backend service for the request

Backend Service



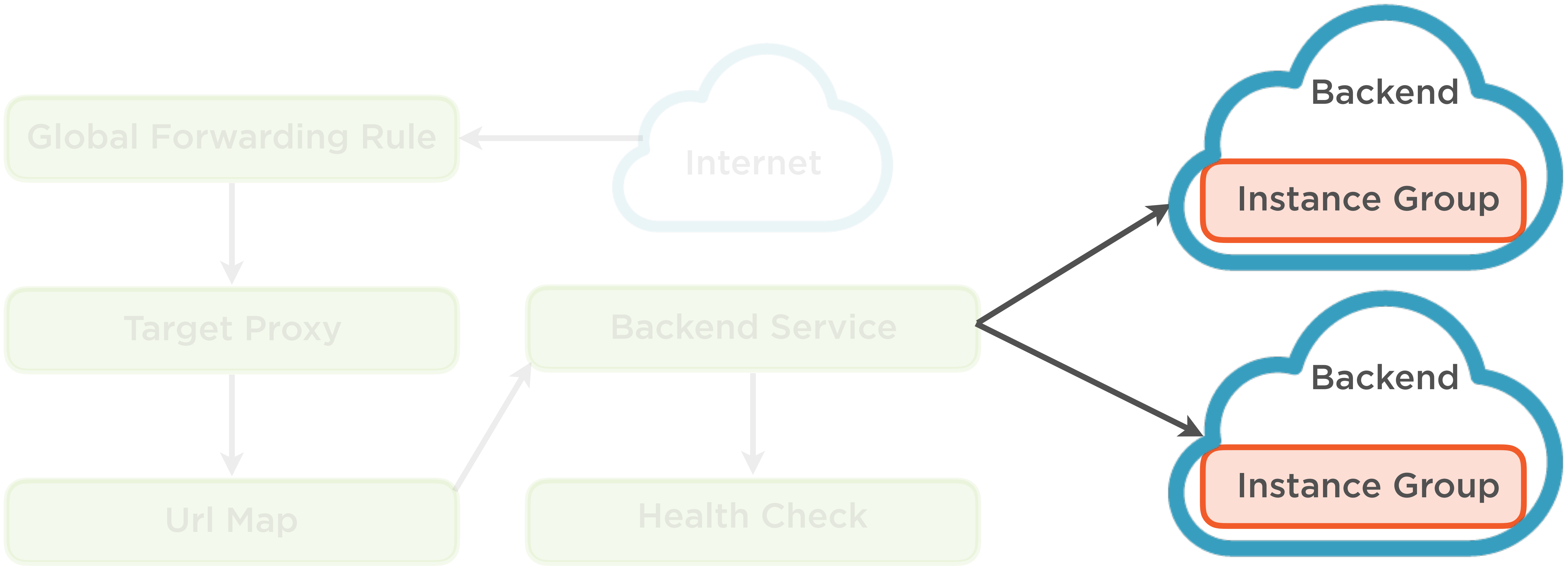
The backend service directs each request to an appropriate backend based on serving capacity, zone, and instance health of its attached backends

Health Check



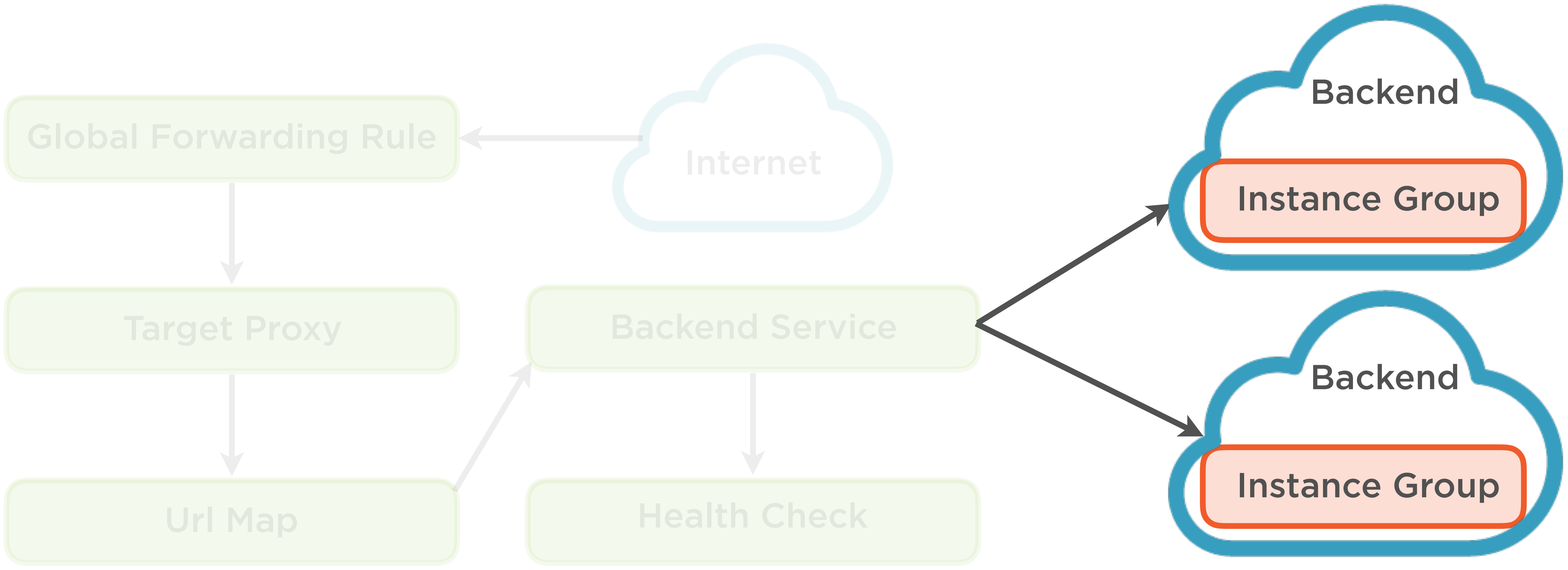
The health of each backend instance is verified using either an HTTP health check or an HTTPS health check - if HTTPS, request is encrypted

Traffic Distribution to Backends



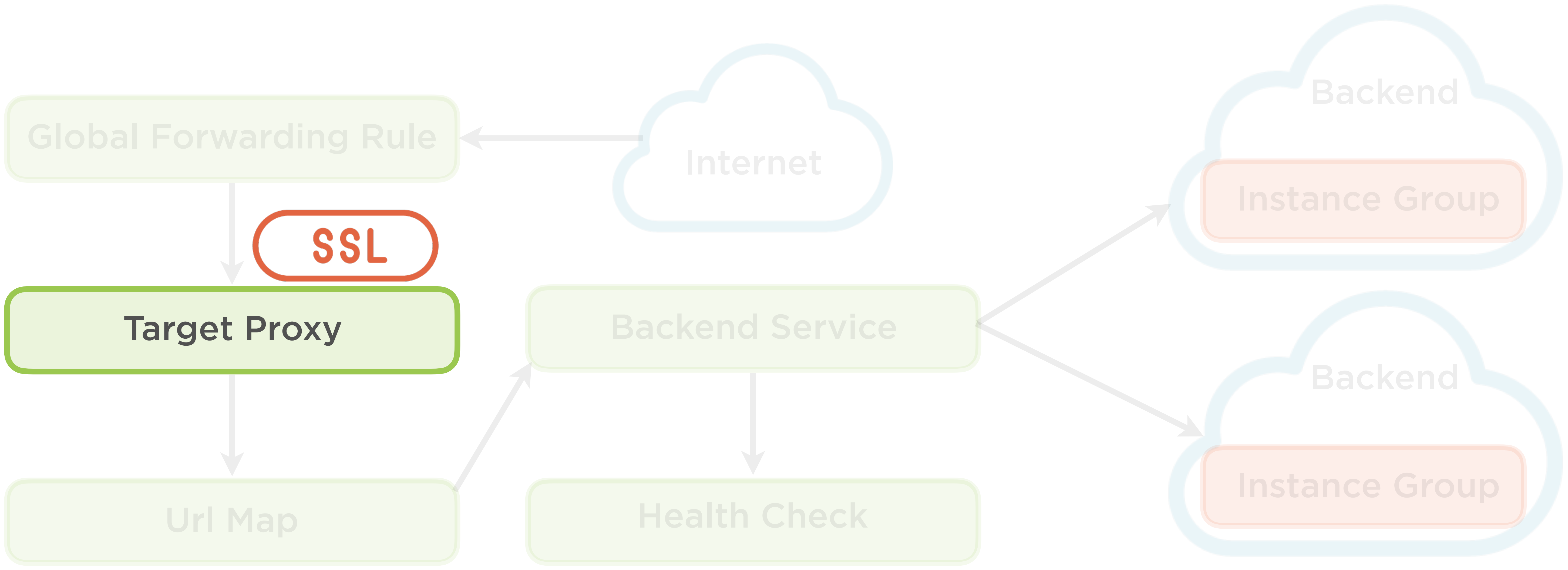
Actual request distribution can happen based on CPU utilization, requests per instance

Autoscaling and Autohealing



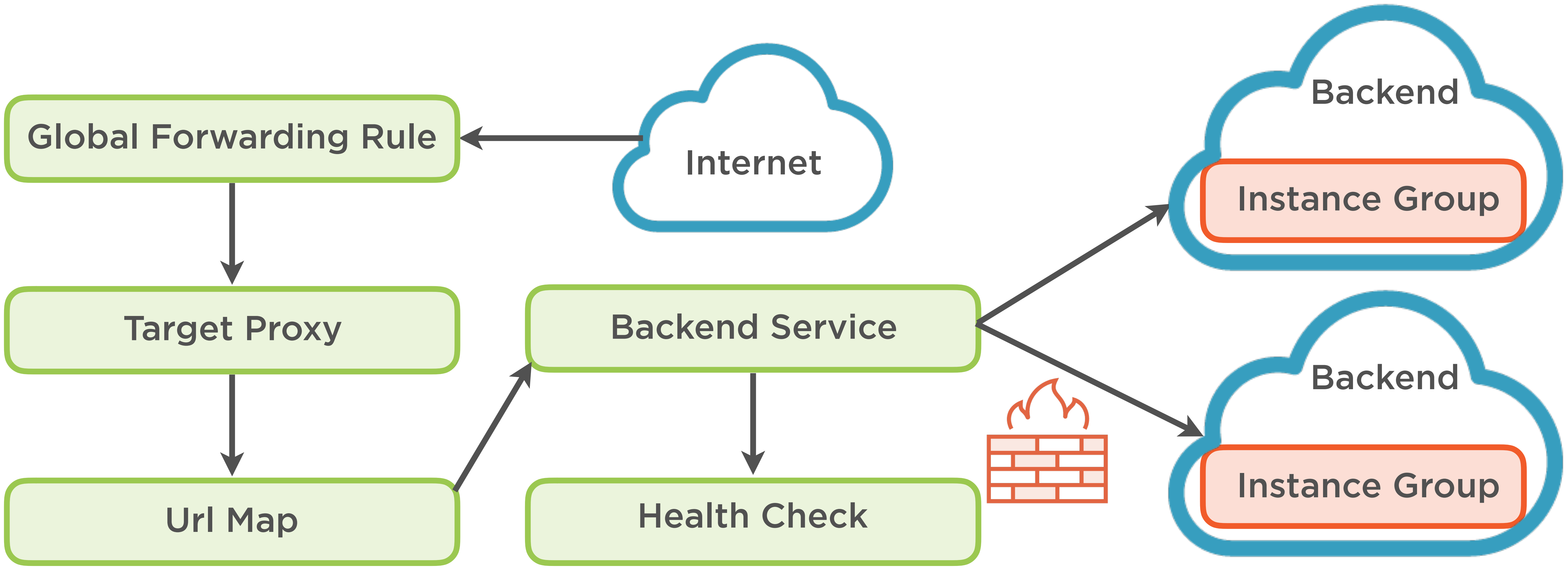
If backends are managed instance groups instances can scale as the traffic scales

SSL Termination



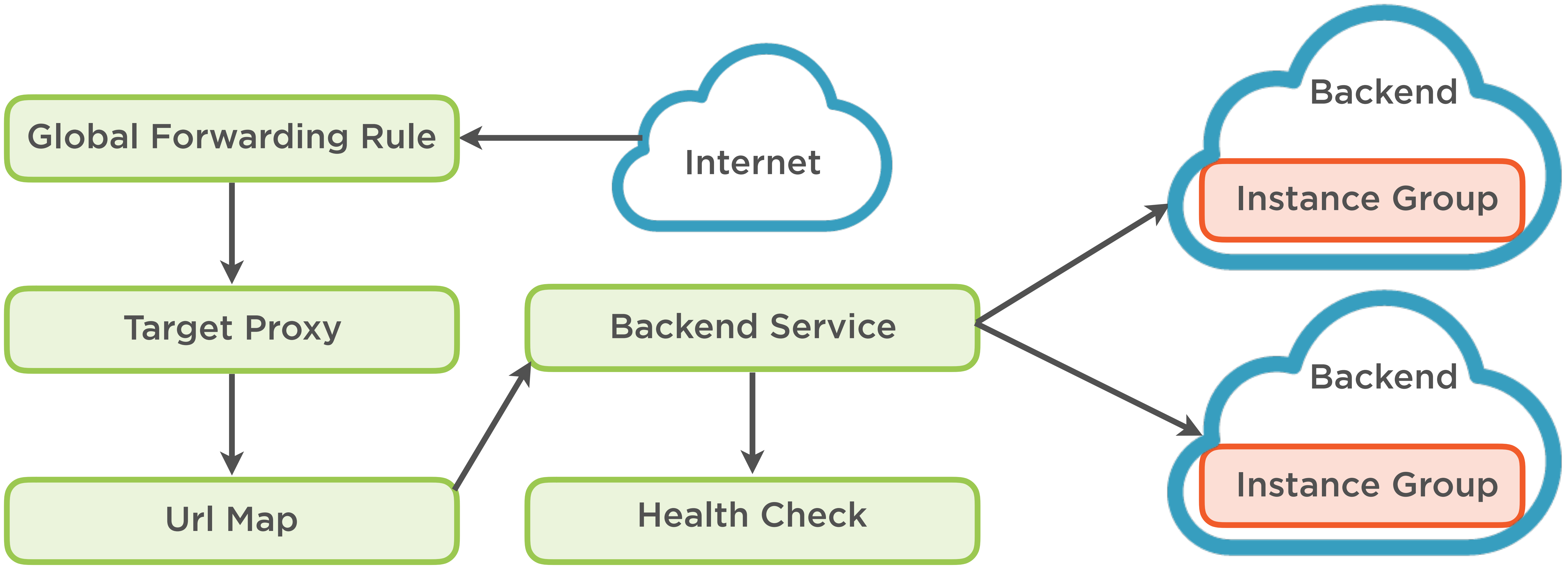
HTTPS load balancing requires the target proxy to have a signed certificate to terminate the SSL connection

Firewall Rules



Must create firewall rules to allow requests from load balancer and health checker to get through to the instances

HTTP(S) Load Balancing



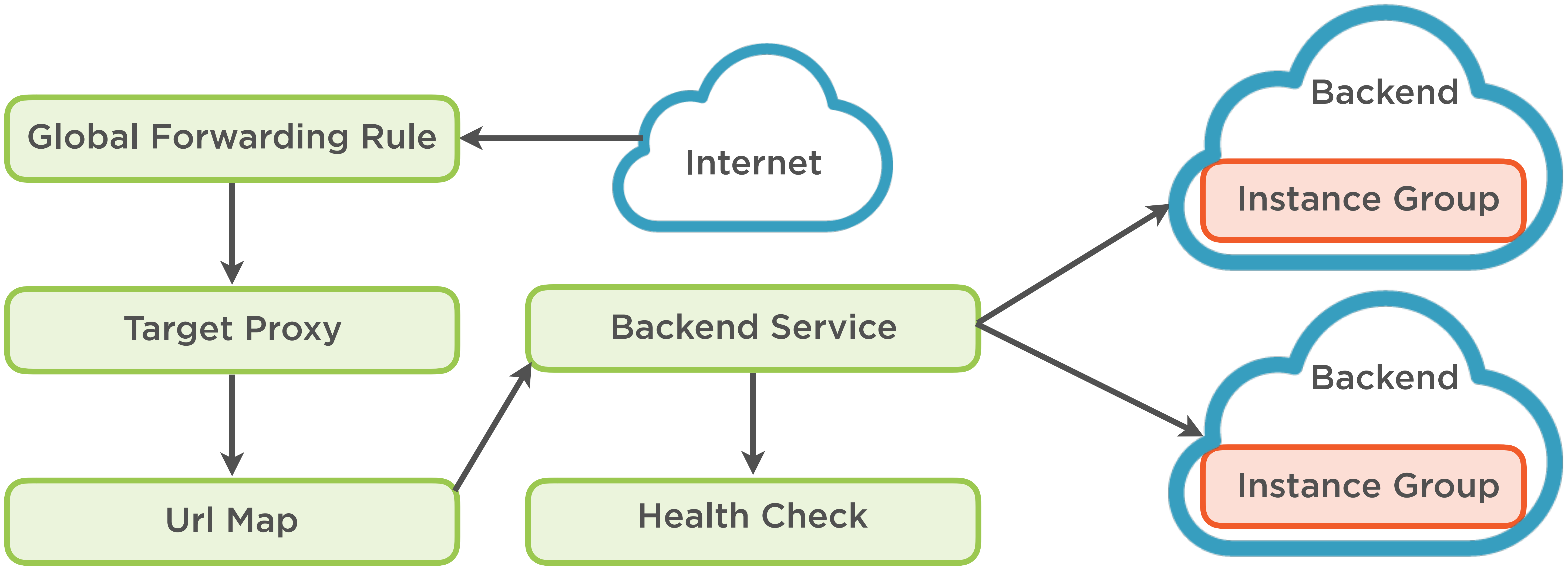
Session Affinity

Session affinity: All requests from same client to same server based on either

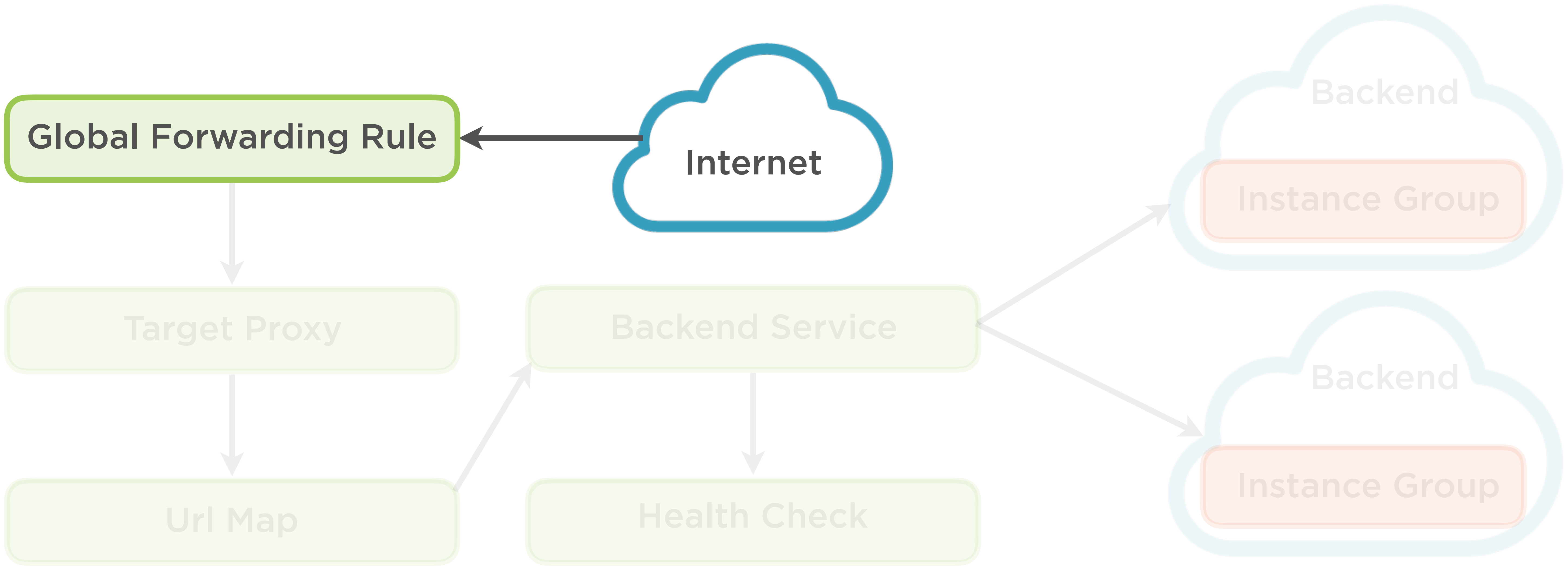
- Client IP
- Cookie

HTTP(S) Load Balancing Components

HTTP(S) Load Balancing



Global Forwarding Rule



Traffic from the internet is sent to a global forwarding rule - this rule determines which proxy the traffic should be directed to

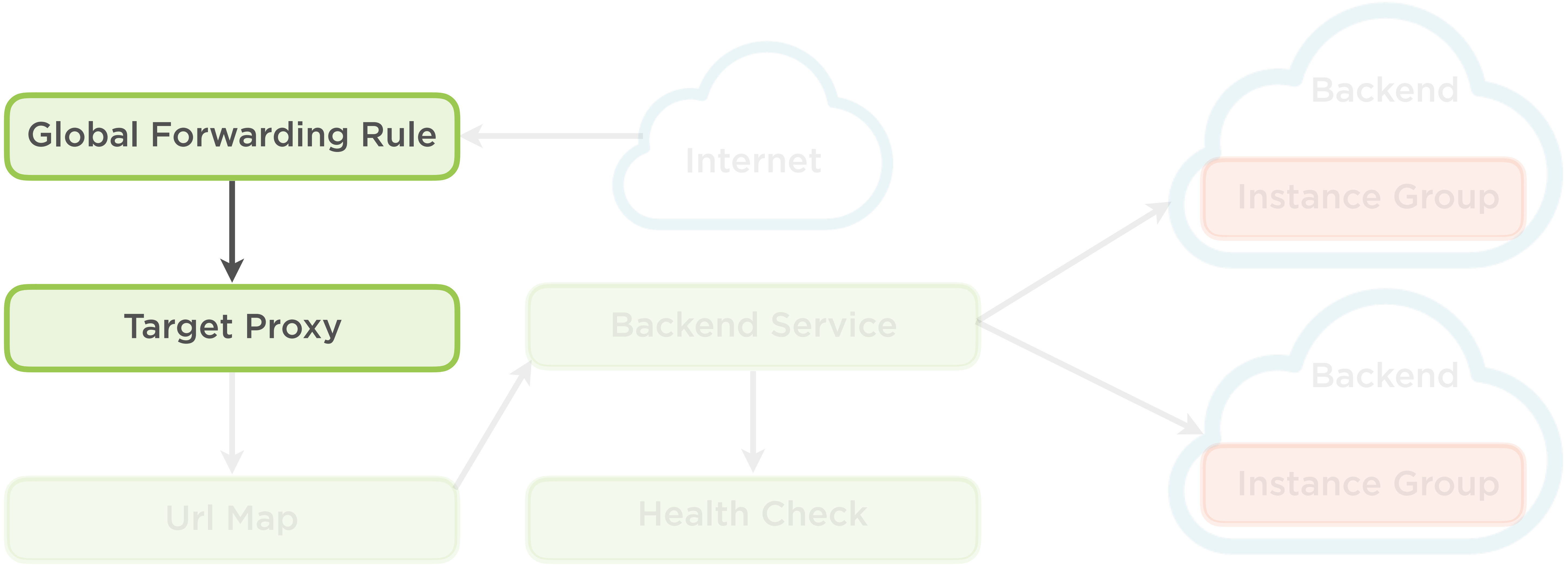
Global Forwarding Rule

Route traffic by IP address, port and protocol to a load balancing proxy

Can only be used with global load balancing HTTP(S), SSL Proxy and TCP Proxy

Regional forwarding rules can be used with regional load balancing and individual instances

Target Proxy



The global forwarding rule directs incoming requests to a target HTTP proxy

Target Proxy

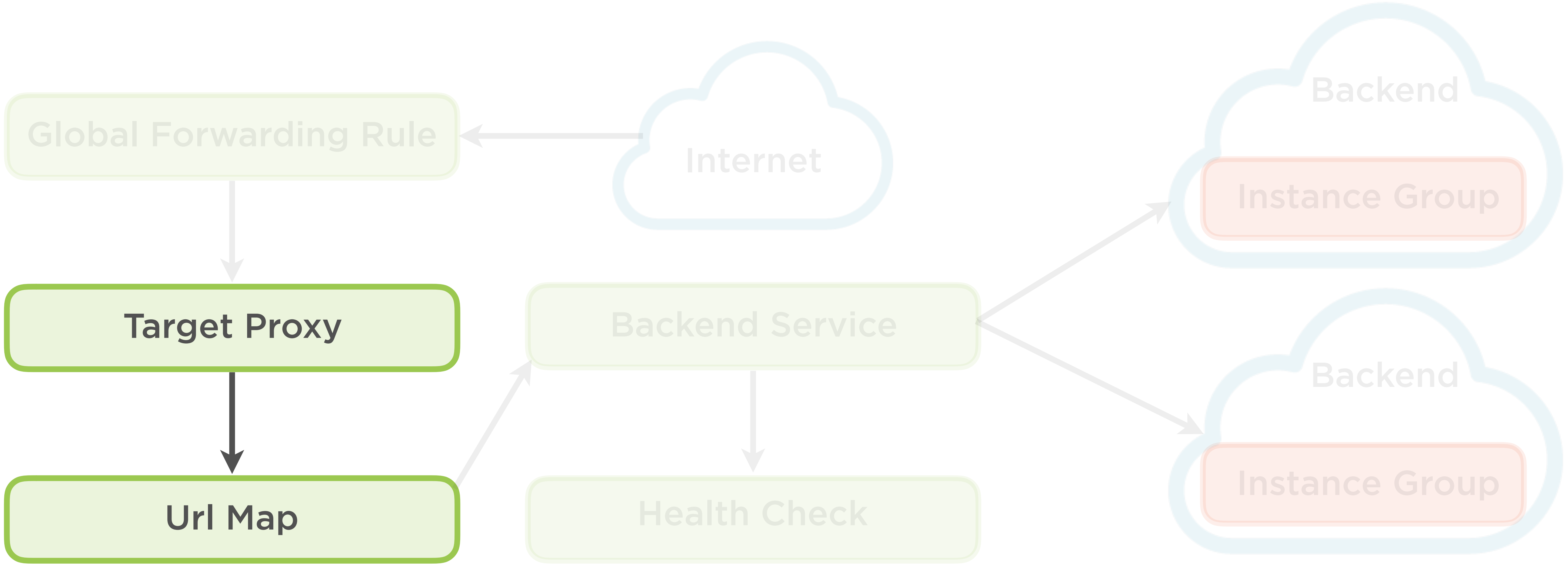
Route the incoming requests to a URL map to determine where they should be sent

Specific to a protocol (HTTP, HTTPS, SSL and TCP)

Should have a SSL certificate if it terminates HTTPS connections

Can connect to backend services via HTTP or HTTPS

URL Map



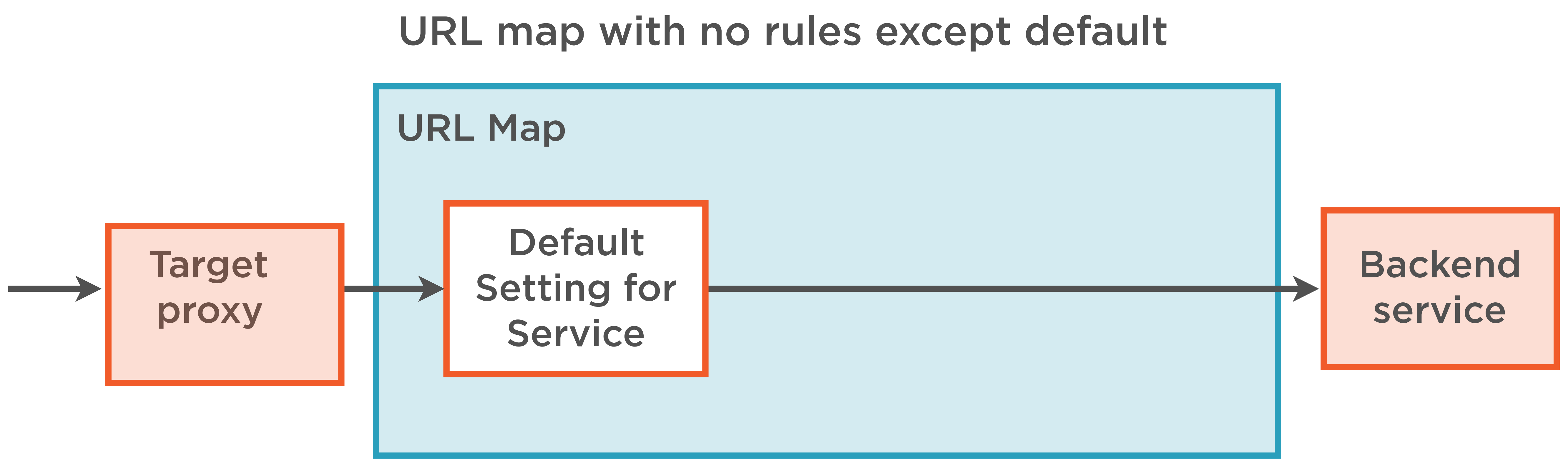
The target HTTP proxy checks each request against a URL map to determine the appropriate backend service for the request

URL Map

Used to direct traffic to different instances based on the incoming URL

- http://www.example.com/audio -> backend service 1
- http://www.example.com/video -> backend service 2

URL Map

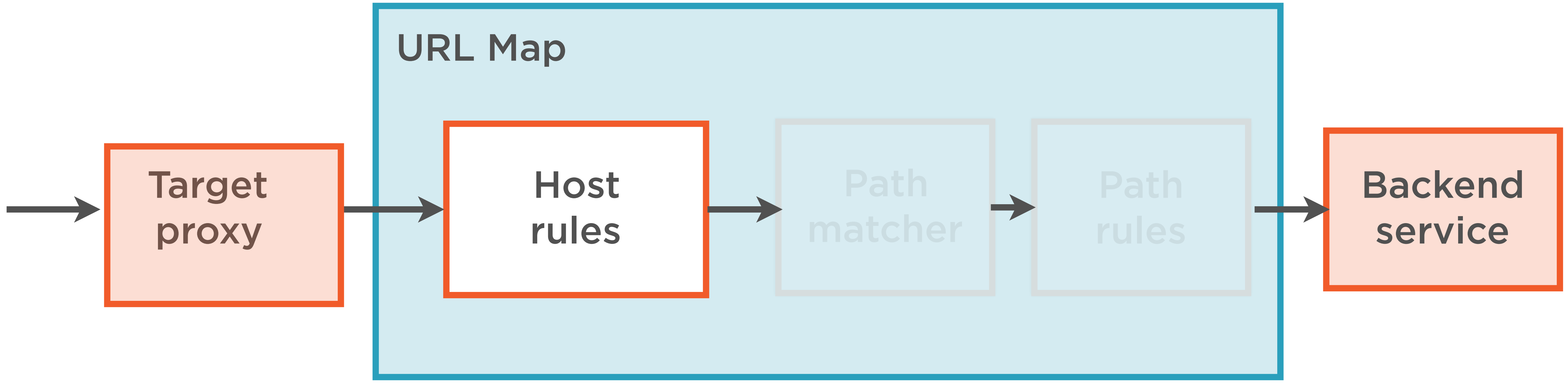

URL map with no rules except default
URL Map
Target proxy
Default Setting for Service
Backend service

All traffic sent to the same groups of instances

Only the /* path matcher is created automatically and directs all traffic to the same backend service

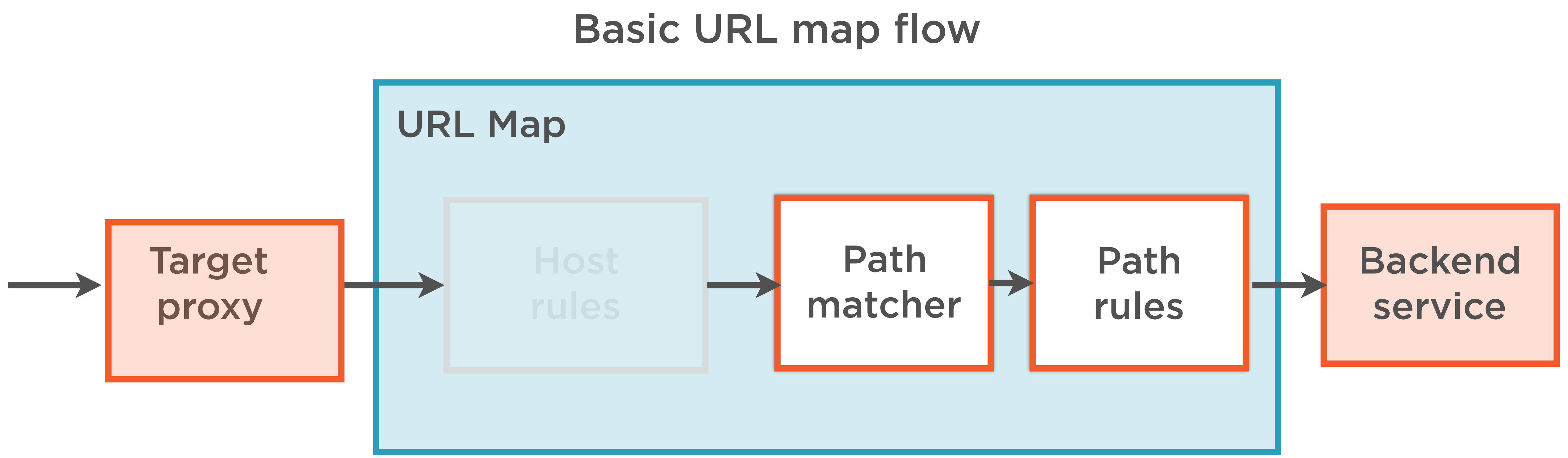
URL Map

Basic URL map flow



Host rules — example.com, customer.com

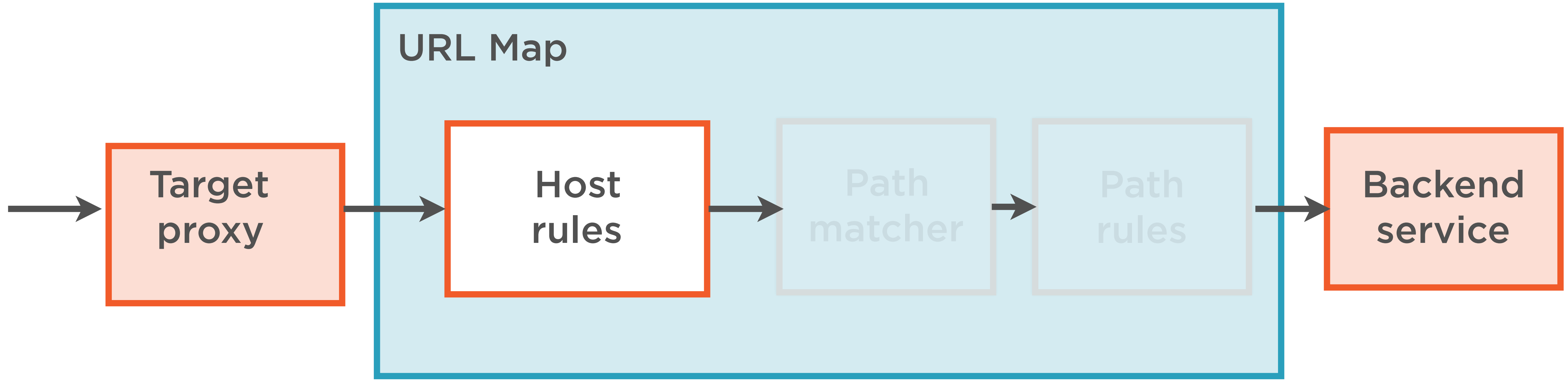
URL Map



Path rules — /video, /video/hd, /video/sd

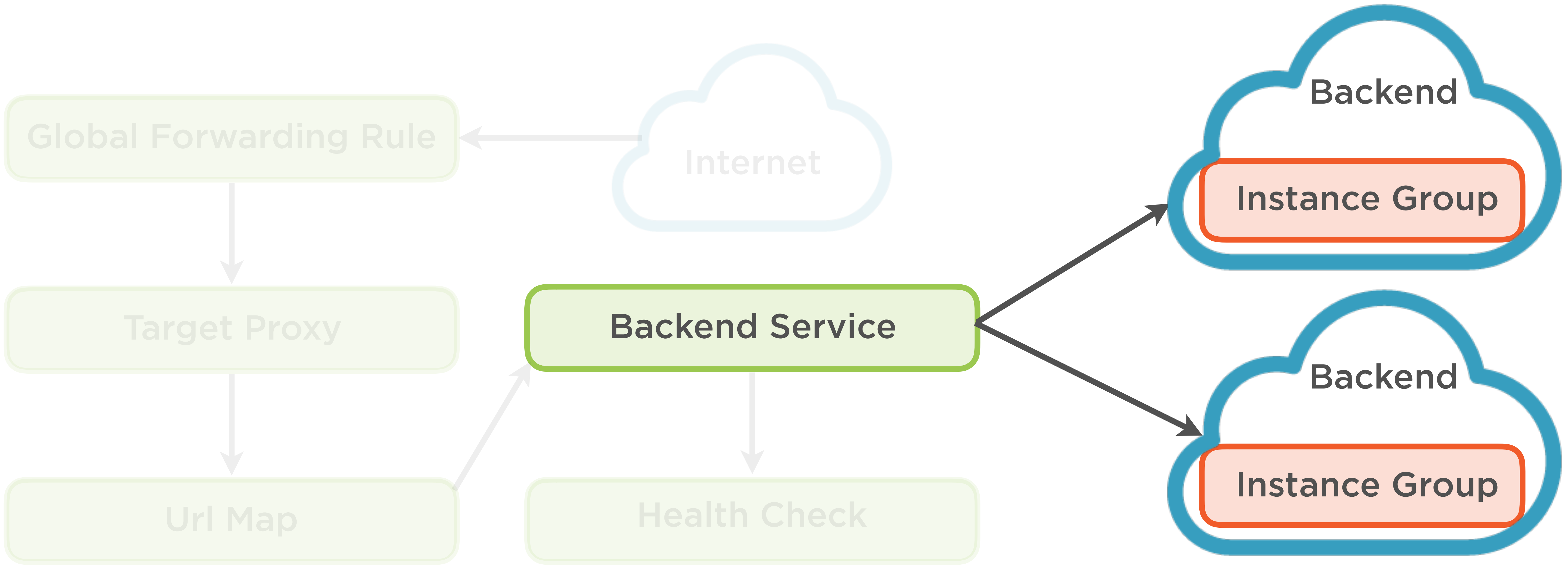
URL Map

Basic URL map flow



A default path matcher /* is created automatically. Traffic which does not match other path rules is sent to this default service

Backend Service



The backend service directs each request to an appropriate backend based on serving capacity, zone, and instance health of its attached backends

Backend Service

Centralized service for managing backends

Backends contain instance groups which handle user requests

Knows which instances it can use, how much traffic they can handle

Monitors the health of backends and does not send traffic to unhealthy instances

Backend Service Components

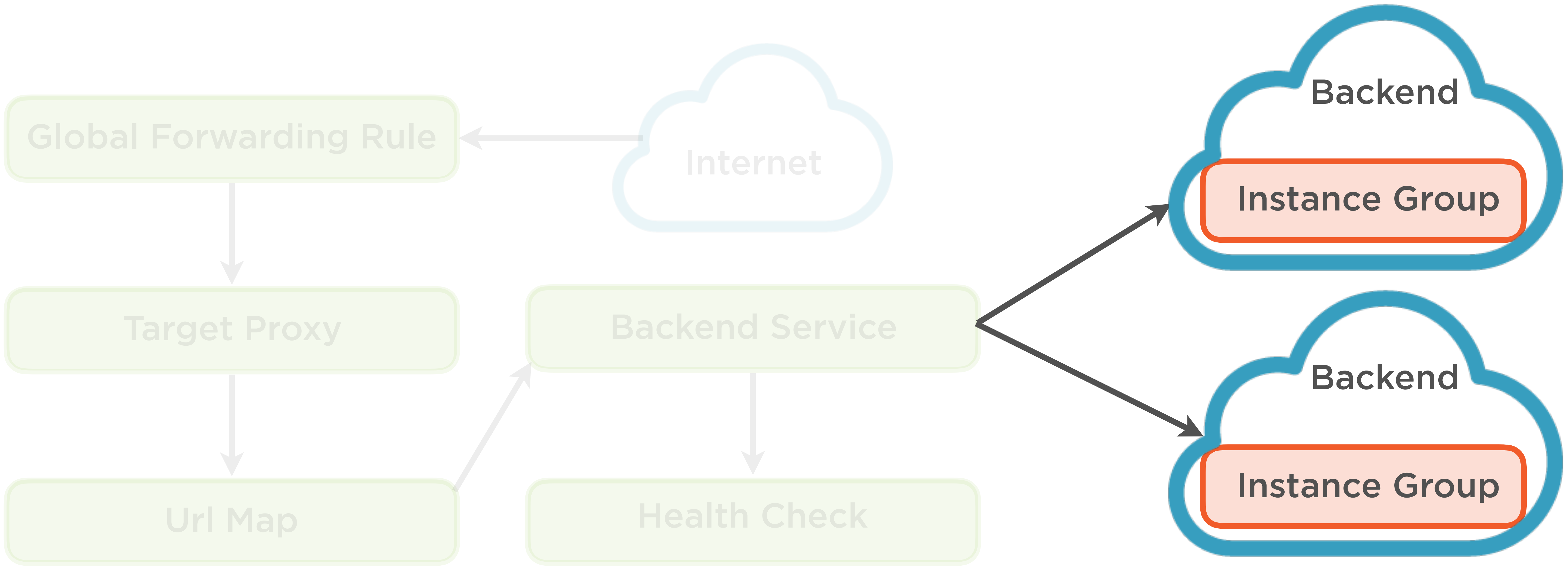
Health Check: Polls instances to determine which one can receive requests

Backends: Instance group of VMs which may be automatically scaled

Session Affinity: Attempts to send requests from the same client to the same VM

Timeout: Time the backend service will wait for a backend to respond

Backends



Actual request distribution can happen based on CPU utilization, requests per instance

Backends

Instance group: Can be a managed or unmanaged instance group

Balancing mode: Determines when the backend is at full usage

- CPU utilization, Requests per second

Capacity setting: A % of the balancing mode which determines the capacity of the backend

Backend Buckets

Allow you to use Cloud Storage buckets with HTTP(S) load balancing

Traffic is directed to the bucket instead of a backend

Useful in load balancing requests to static content

Load Distribution

Uses CPU utilization of the backend or requests per second as the balancing mode

Maximum values can be specified for both

Short bursts of traffic above the limit can occur

Load Distribution

Incoming requests are first sent to the region closest to the user, if that region has capacity

Traffic distributed amongst zone instances based on capacity

Round robin distribution across instances in a zone

Round robin can be overridden by session affinity

Demo

Configure a cross-regional load balancer using unmanaged instance groups

Demo

Configure a cross-regional load balancer using managed instance groups

Load test the load balancers to observe autoscaling and traffic distribution in action

Summary

Introducing instance templates

Managed instance groups and unmanaged instance groups

Architectural overview of the HTTP(S) load balancer and its components

Cross-regional load balancers with unmanaged and managed instance groups

Autoscaling with managed instance groups