ambassador
EDGE STACK

Implementing Ambassador Edge Stack

A Guide for Engineering Leaders

Trusted by developers at

NVIDIA. Microsoft ptc ticketmaster® Chick-fil-A

How to Implement Ambassador Edge Stack to Modernize Your System and Save You Money

Your team needs an API gateway to handle traffic coming into your Kubernetes clusters, and you're considering Ambassador Edge Stack. You need to consider two big-picture things as your team moves forward: personnel and budget. How do teams use Edge Stack? Who owns it? And what does it cost to run? How many CPUs is this thing going to require?

> *Ambassador makes it very easy for us to manage endpoints across all our regions worldwide and is able to seamlessly adapt and work with every region's 80 different endpoints, each with varying configuration requirements.*

Nashon Steffen

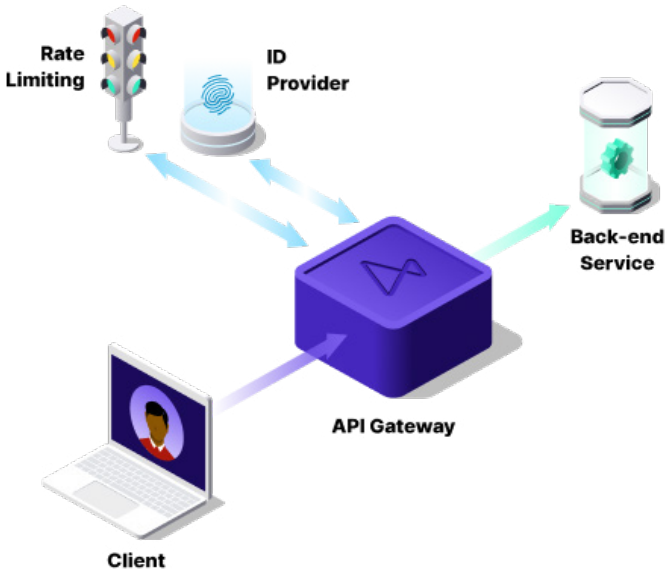
Staff Infrastructure Development Engineer

Mercedes-Benz

Architecture — Designed and Built for Cloud Native Workloads

Let's start by looking at the architecture of Edge Stack and where it sits in your larger environment, as that influences how teams use it. Ambassador Edge Stack is a single entry point for traffic into your Kubernetes cluster. Multiple Edge Stack installations can co-exist in one cluster, but this is not commonly necessary. The architecture of Edge Stack is comprised of:

- Load Balancer - created by your cloud provider or, if you are running on your servers, your team, as part of the installation process
- Edge Stack proxy - the containers that route traffic to your internal services
- Redis - used for rate limiting and authentication
- Ambassador Agent - communicates with Ambassador Cloud to provide your team with information via our dashboards and service catalog, and to verify your license and usage.



Teams install Edge Stack resources into an "ambassador" namespace, and the installation permits the resources to read routing configuration resources (Mappings) from the rest of the cluster. This design strikes a balance between distributed flexibility and centralized control. It allows shared network infrastructure to be used by many different and non-coordinating teams, all bound by the policies and constraints set by cluster operators.

How Edge Stack Enables Teams to Deploy with Speed and Safety

One of the biggest problems that Edge Stack solves for organizations is allowing development teams to quickly deploy and make changes without being blocked by a centralized cluster operations team (this team may have many different names, i.e., DevOps/Platform/Cloud Engineering, etc). It allows application teams to own their routing configuration, and they only have to understand one Kubernetes resource, the Mapping, to get traffic flowing to their apps. Cluster operators have their workload reduced by not being the gatekeepers of routing configuration. We commonly see responsibilities divided as follows:

Platform Team

- Installs, updates, and maintains Edge Stack
- Sets up dashboards for monitoring Edge Stack resource usage, health, and logs
- Configures external authentication services like SSO or JWT validation
- Sets global rate limit or authentication requirements
- Creates Host resources (what hostnames Edge Stack will listen on)
- Configures TLS settings (using the built-in Let's Encrypt integration or using another certificate-creation system)

Developer Teams

- Create Mappings to route traffic to their applications
- Control retry and timeout settings
- Set app or route-specific rate limiting and authentication
- Set up app-specific dashboards using metrics from Edge Stack

We can help you to:

- Identify which team will own the Edge Stack installation
- Train your teams on their roles and how to succeed with their responsibilities

Required Resources — Edge Stack Benchmarks

You now know the people you need to run Edge Stack. What about compute resources?

The Edge Stack gateway application is a wrapper around Envoy Proxy that retrieves mappings, compiles them into an Envoy configuration, and manages rate limiting and authentication requests. Envoy is extremely fast, but every time a request is received, Envoy must search its configuration for where to route the request. Because of that, Edge Stack benefits from being on CPU-optimized servers with faster processor speeds. Performance depends on the number and complexity of three different parameters:

- Hosts (hostnames Edge Stack might receive traffic from)
- Mappings (routing configurations)
- Backends (services or containers the mappings point to)

The range of possible configurations is massive, so it is difficult to provide exact performance numbers upfront. You might have 1,000 Hosts pointing to ten Mappings that route traffic to one backend, or you might have one Host that points to 10,000 Mappings pointing to 10,000 backends, and anything between those scenarios. Our team can help you determine your situation during the initial implementation process. However, we can provide some general benchmark numbers to give you a starting place for budgeting the hardware to allocate (pricing dependent on your vendor or data center).