envoycon

NORTH AMERICA

envoycon
NORTH AMERICA

Extending Envoy with Web Assembly

Daneyon Hansen
Software Engineer, Tetrate

Envoy Extensibility

- Supports a variety of extension points.
- Extensibility options:
    - **Native C++ Filters**
    - **Lua-based Filters**
    - **WebAssembly (Wasm) Filters**

Introduction

webassemby.org states:

> “WebAssembly (Wasm) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.”

Introduction

webassemby.org states:

> “WebAssembly (Wasm) is a **binary instruction format** for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.”

Introduction

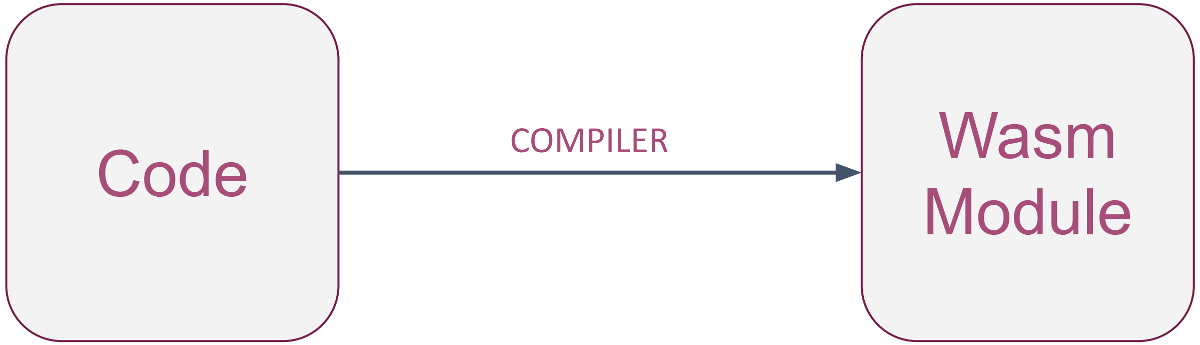
webassemby.org states:

> “WebAssembly (Wasm) is a binary instruction format for a **stack-based virtual machine**. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.”

Introduction

webassemby.org states:

> “WebAssembly (Wasm) is a binary instruction format for a stack-based virtual machine. Wasm is designed as **a portable compilation target** for programming languages, enabling deployment on the web for client and server applications.”

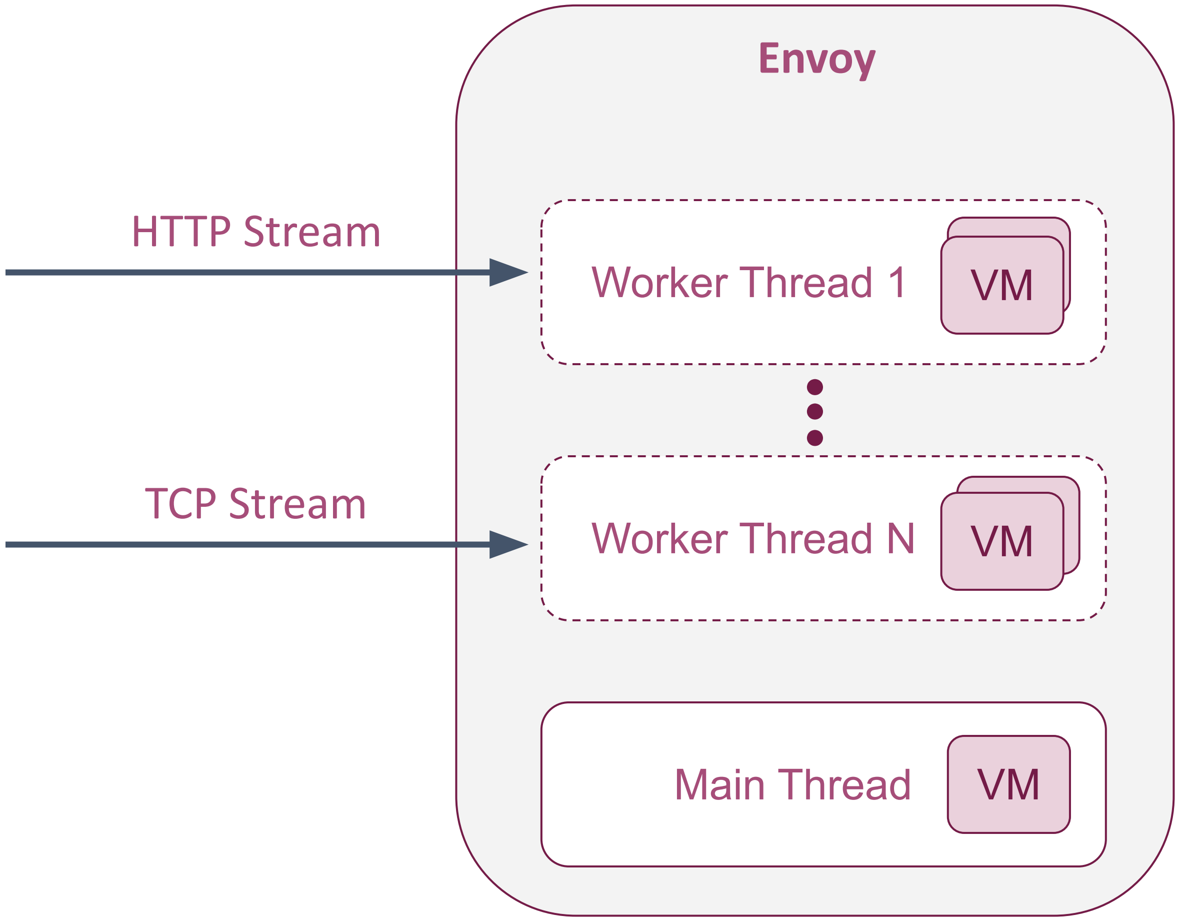


Wasm Benefits

- **Developer friendly**: Support for multiple languages.
- **Fast:** Encoded in an efficient binary format.
- **Safe:** Code is validated and executed in a memory-safe, sandboxed environment.
- **Portable:** Can be embedded in browsers, run as a stand-alone VM, or integrated in other environments.

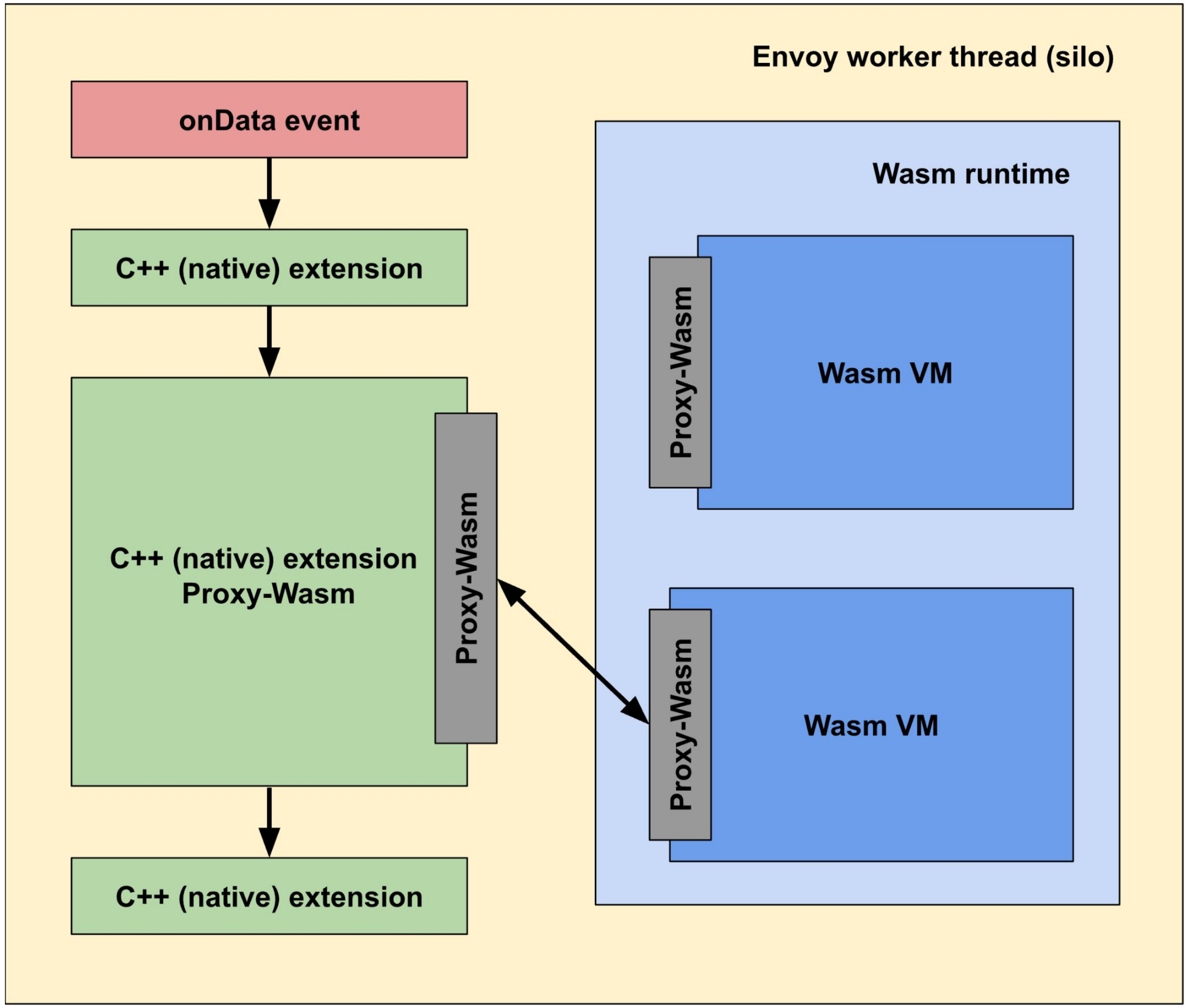
Wasm in Envoy

- Supports Wasm runtimes
- Multi-threaded model
- Threads are independent
- Wasm VM = a loaded .wasm module



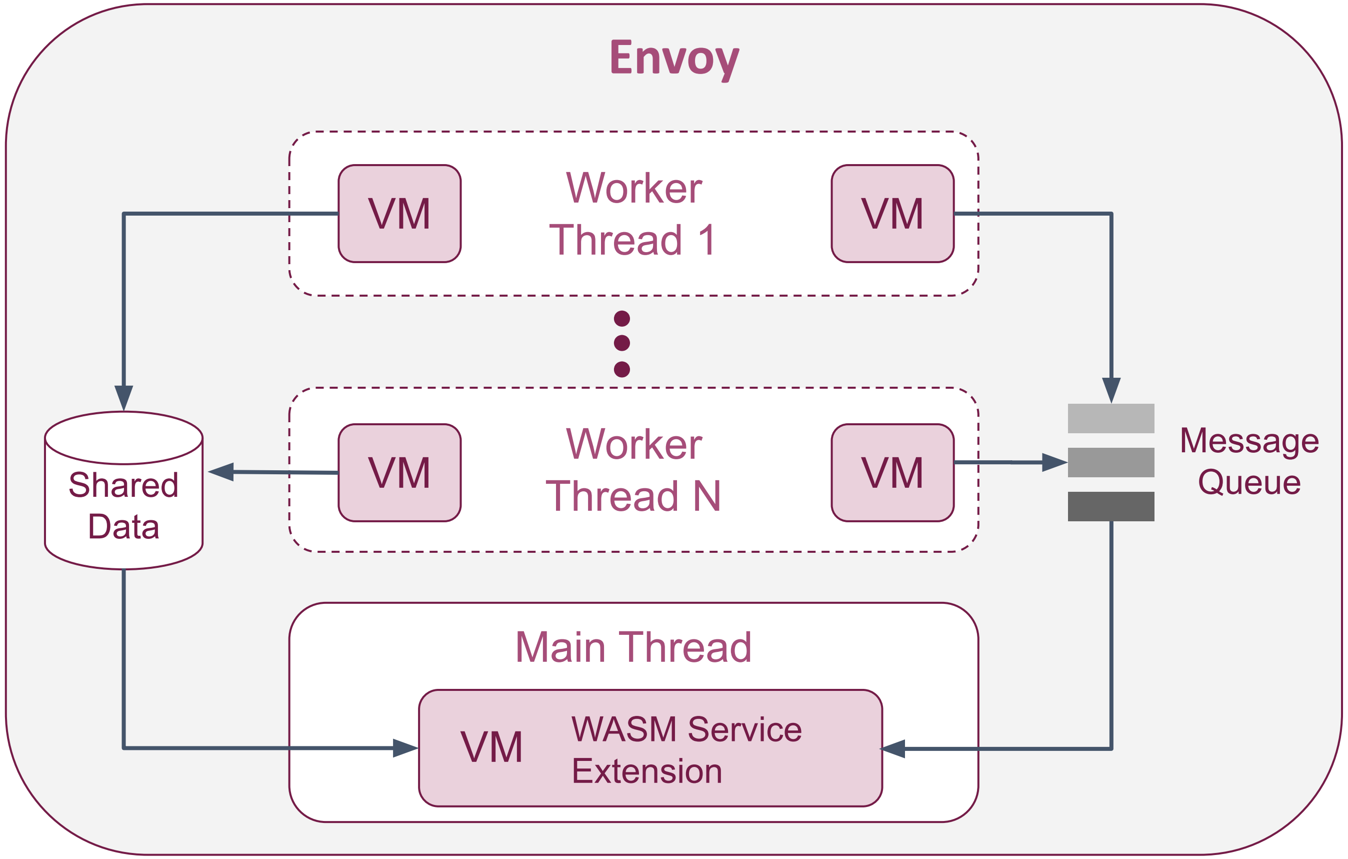
Proxy-Wasm

- Implementation-agnostic interfaces for running Wasm in proxies.
- Implemented by Envoy
- Define interactions between Envoy<>VM
- SDKs



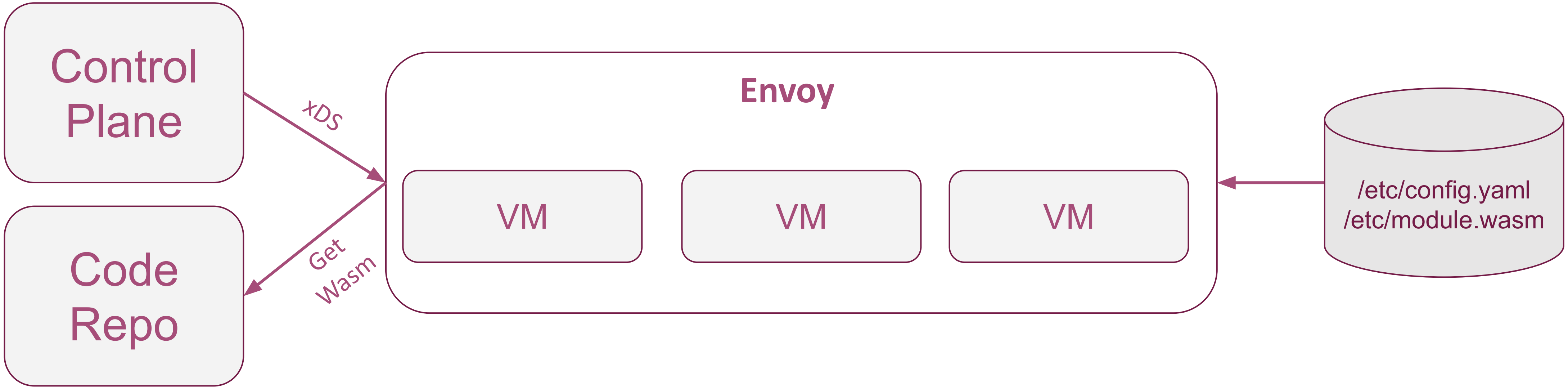
Wasm Service

- Stateless
    - HTTP filters
    - Network filters
- Stateful
    - Wasm service
- Supported APIs:
    - Stats/Log
    - Shared Data
    - Message Queue
    - Timer
    - gRPC/HTTP

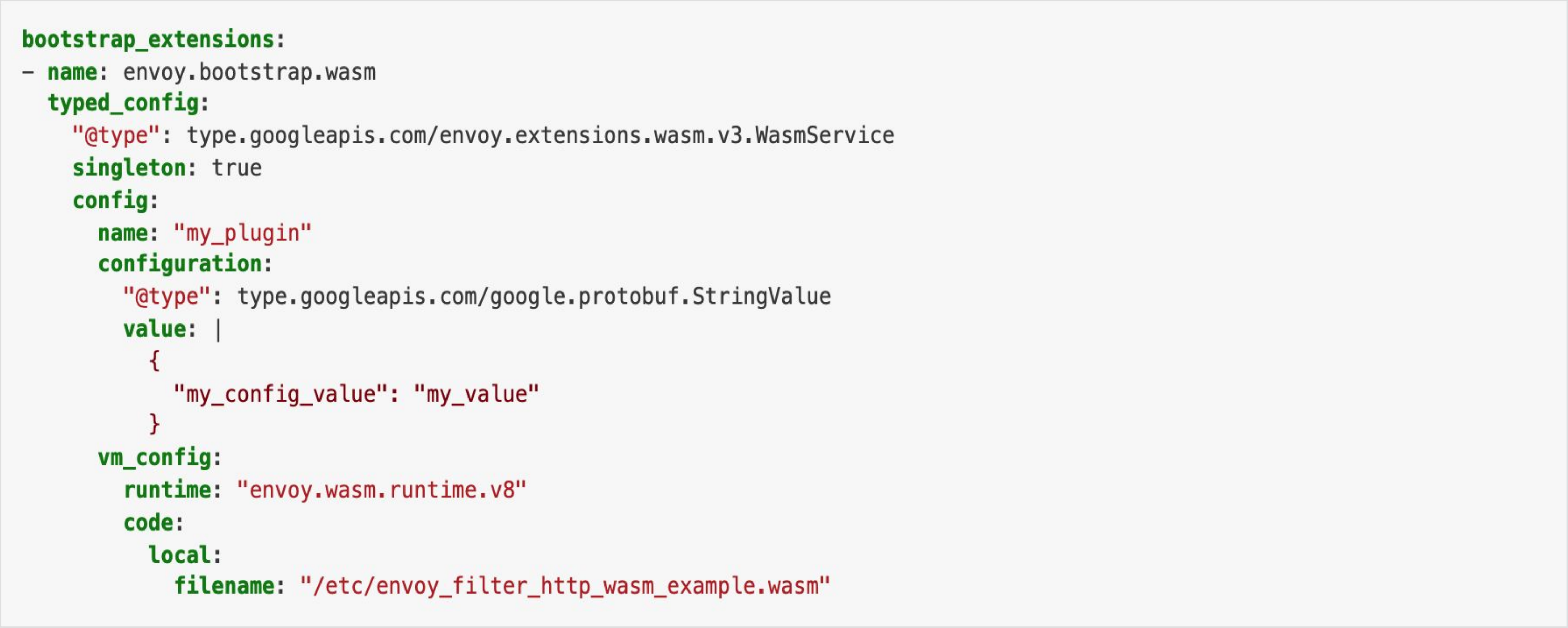


Envoy Wasm Configuration

envoycon
NORTH AMERICA

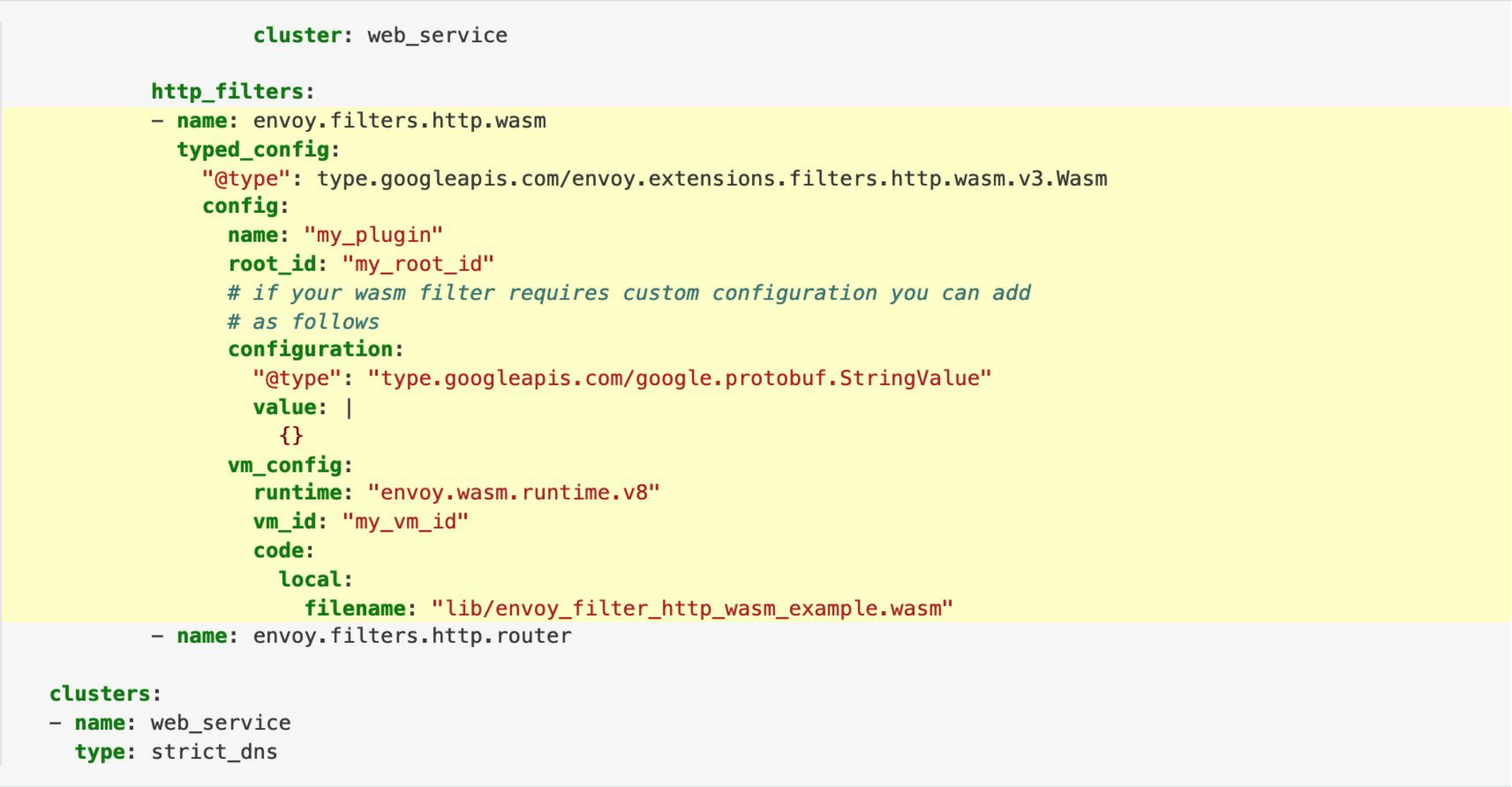


Wasm Service Config



```yaml
bootstrap_extensions:
- name: envoy.bootstrap.wasm
  typed_config:
    "@type": type.googleapis.com/envoy.extensions.wasm.v3.WasmService
    singleton: true
    config:
      name: "my_plugin"
      configuration:
        "@type": type.googleapis.com/google.protobuf.StringValue
        value: |
          {
            "my_config_value": "my_value"
          }
      vm_config:
        runtime: "envoy.wasm.runtime.v8"
        code:
          local:
            filename: "/etc/envoy_filter_http_wasm_example.wasm"
```

Wasm HTTP Filter Config



```yaml
                cluster: web_service

        http_filters:
        - name: envoy.filters.http.wasm
          typed_config:
            "@type": type.googleapis.com/envoy.extensions.filters.http.wasm.v3.Wasm
            config:
              name: "my_plugin"
              root_id: "my_root_id"
              # if your wasm filter requires custom configuration you can add
              # as follows
              configuration:
                "@type": "type.googleapis.com/google.protobuf.StringValue"
                value: |
                  {}
              vm_config:
                runtime: "envoy.wasm.runtime.v8"
                vm_id: "my_vm_id"
                code:
                  local:
                    filename: "lib/envoy_filter_http_wasm_example.wasm"
        - name: envoy.filters.http.router

clusters:
- name: web_service
  type: strict_dns
```

Demo

https://github.com/danehans/envoycon

References

- WebAssembly: https://webassembly.org/
- Proxy-wasm: https://github.com/proxy-wasm/spec
- Proxy-wasm-go-sdk: https://github.com/tetratelabs/proxy-wasm-go-sdk
- Distributing Wasm Modules using Istio: https://istio.io/latest/docs/ops/configuration/extensibility/wasm-module-distribution/
- Istio Wasm Extensions API Design Spec: https://docs.google.com/document/d/15GQnhOpLfvgyOZ8VDF5KD-q2cjXEMLdTQMm0YPswIu0/edit?usp=sharing

Acknowledgements

- John Plevyak
- Mrunmayi Dhume
- Michael Cieplak
- Peter Jausovec