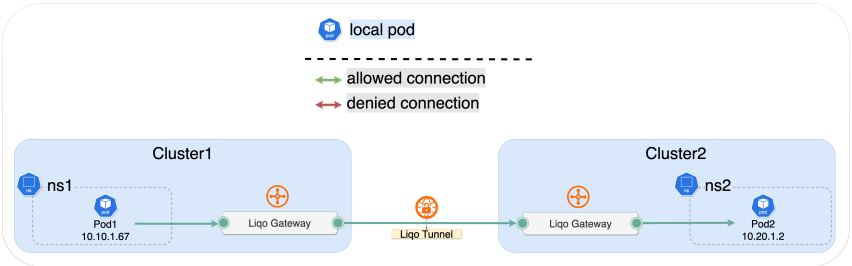
# First Steps in Liqo Security

## NFTables-based Policies for Liqo

**Giulio Brazzo**

# Table of Contents

- `offloading`: transparently extend the local cluster by offloading workloads to remote clusters.



- **NFTables**: framework for packet filtering and classification in the Linux kernel, successor of iptables.

```
table inet mytable {
    chain mychain {
        type filter hook input priority 0; policy drop;
        # Some rules here
    }
}
```

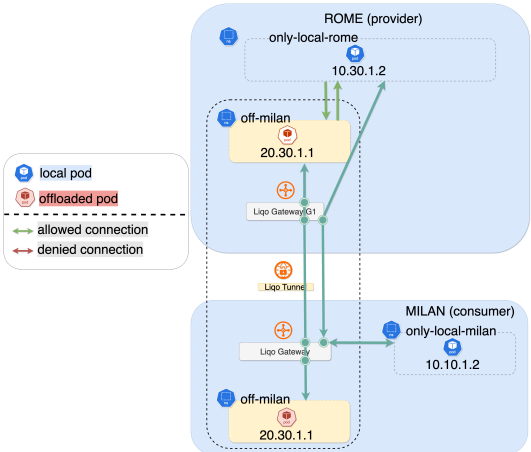
- **NAT + route**: Currently Liqo uses NFTables for NATting and routing, but no traffic filtering.

# Table of Contents

2 Current behaviour

# Inter-cluster traffic

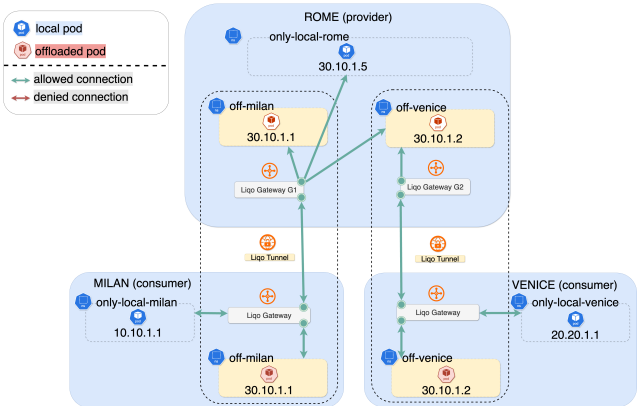
## 2 Current behaviour




- Liqo enables seamless communication between pods across clusters.
- By default, inter-cluster traffic is unrestricted.

# Inter-cluster traffic with multiple consumers

## 2 Current behaviour




- Kubernetes allows unrestricted communication between pods within the same cluster by default.

# Table of Contents

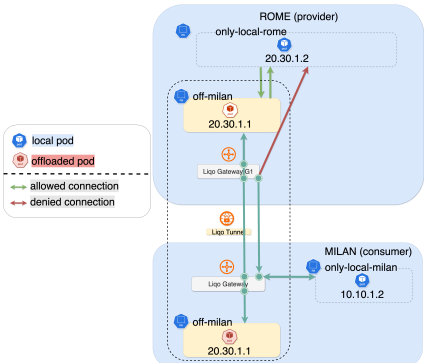
3 Problem statement

# Security Scenarios

- **Provider Protection**: Isolate tenants in multi-tenant environments.
  - Single or multiple consumer clusters offloading workloads to shared provider clusters.
- **Consumer Protection**: Prevent unauthorized access to consumer cluster resources.

# Single consumer provider protection example

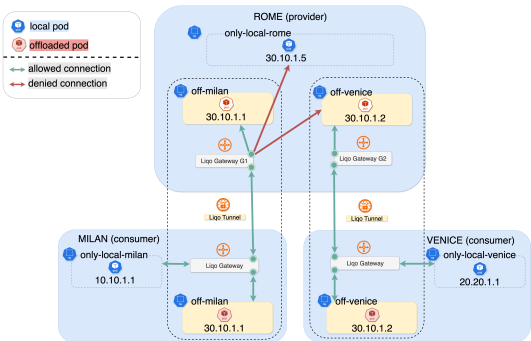
## 3 Problem statement




- Consumer cluster offloads workloads to a provider cluster.
- NFTables restrict traffic only to specific offloaded pods.

# Multiple consumers provider protection example

## 3 Problem statement




- Multiple consumer clusters offload workloads to the same provider.
- Each consumer cluster accesses only its own resources.
- NFTables rules enforce isolation between consumers.

# Table of Contents

4 Implementation

# Modified Custom Resource Definitions (CRDs)

- `FirewallConfiguration`: Defines nftables rules applied at the gateway level.
- Rules specify action (accept/drop), IP ranges, and counters for monitoring.
- CRDs enable declarative configuration of firewall policies integrated in the Liqo workflow.

# CRD FirewallConfiguration Example

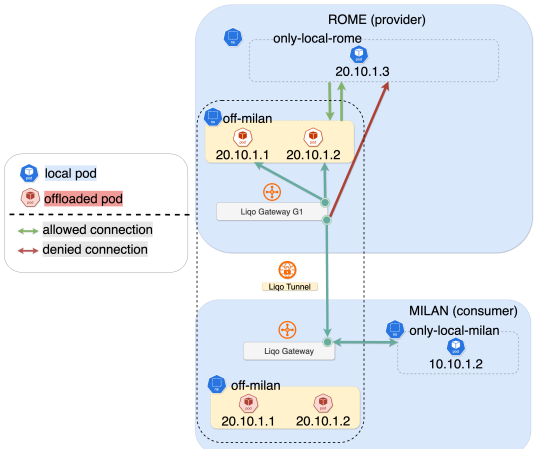
4 Implementation

```yaml
kind: FirewallConfiguration
metadata:
  labels:
    liqo.io/firewall-category: secureGateway
spec:
  table:
    name: liqo-table
    family: IPv4
    chains:
      - name: liqo-chain
        hook: forward
        priority: 0
        policy: drop
        type: filter
        rules:
          filterRules:
            - name: allow_traffic
              action: accept
              counter: true
              match:
                ip:
                  value: 10.0.0.0-10.0.0.255
                  position: src
                op: eq
```

- `FirewallConfiguration` CRD defines firewall rules for offloaded pods.
- **Action**: Specifies whether to accept or drop traffic.
  - Accept: Allow traffic to the specified pods.
  - Drop: Block traffic to the specified pods.
- **IP Ranges**: Defines the source and destination IP ranges for the rules.
- **Counter**: Tracks the number of packets and bytes matching the rule.

# Single Consumer Example

## 4 Implementation



# FirewallConfiguration → nftables

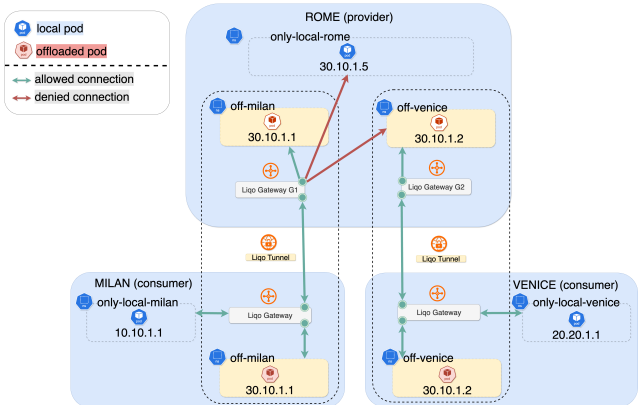


```
table ip liqo-table {
  chain liqo-chain {
    type filter hook forward priority 0; policy drop;

    ip daddr 20.10.1.1 counter accept
    ip daddr 20.10.1.2 counter accept
  }
}
```

# Multiple Consumer Example

## 4 Implementation



# FirewallConfiguration → nftables



```
table ip liqo-table {
  chain liqo-chain {
    type filter hook forward priority 0; policy drop;

    ip daddr 30.10.1.1 counter accept
  }
}
```

# Webhook Modifications

4 Implementation

- Admission webhook extended to validate and mutate requests.
- Various checks performed:
  - Validate CRD presence and correctness.
  - Check for conflicts in IP ranges.
  - Added checks for ”counter” field validity and ”action” field correctness.

# Possible implementation for automatically installing rules



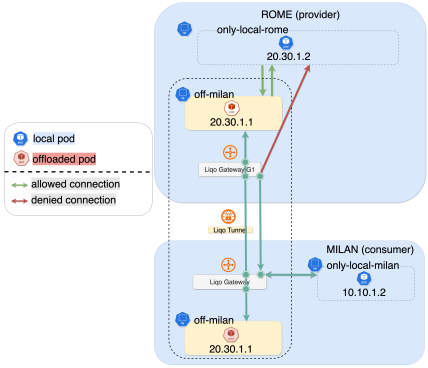
- At peering time, nftables rules are automatically installed on the gateway nodes.
- Rules enforce strict access control:
  - Allow traffic only between authorized offloaded pods.
  - Block unauthorized inter-cluster connections.
- During reconcile, rules are updated to reflect cluster state changes, ensuring up-to-date security.

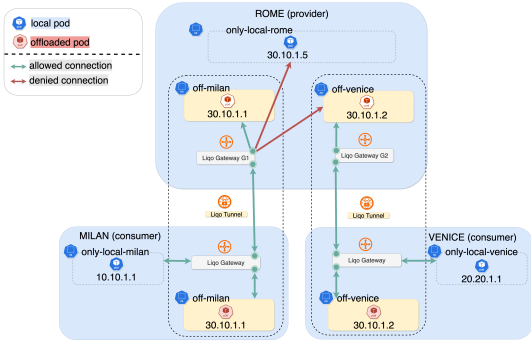
# Table of Contents

5 Conclusions

# Final Architecture

## 5 Conclusions



# First Steps in Liqo Security

*Thank you for listening!*