Tigera / Calico – Update

Mar 2021
Tigera: Leader in Kubernetes Security & Observability

- Inventor and maintainer of Project Calico
  - Most adopted open source Kubernetes networking & security solution
  - 1+ Million nodes in 166 countries
- eBPF, Standard Linux, Windows
  - Pluggable data plane
- Envoy
- Customers include
  - Fortune 500 companies
  - Premier financial services institutions
  - Cloud-native companies
  - Telcos and eCommerce giants

Customers

Partners

© 2021 Tigera, Inc. Proprietary and Confidential
Industry Standard for Kubernetes Security

Outcomes

- Scale
- Performance
- Resource Utilization

Invisible

“It just works!”
Continuing to grow our adoption

85% (YoY) growth in number of clusters.

Source: Datadog, Container Report 2020
Third-party CNI **Benchmark** Tests

<table>
<thead>
<tr>
<th>CNI Benchmark August 2020 infraBuilder</th>
<th>Config</th>
<th>Performances (bandwidth)</th>
<th>Resources consumption (cpu/ram)</th>
<th>Security features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MTU setting</td>
<td>Pod to Pod</td>
<td>Pod to Service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP</td>
<td>UDP</td>
<td>TCP</td>
</tr>
<tr>
<td>Antrea</td>
<td>auto</td>
<td>Very fast</td>
<td>Very fast</td>
<td>Very fast</td>
</tr>
<tr>
<td>Calico</td>
<td>manual</td>
<td>Very fast</td>
<td>Very fast</td>
<td>Very fast</td>
</tr>
<tr>
<td>Canal</td>
<td>manual</td>
<td>Very fast</td>
<td>Very fast</td>
<td>Very fast</td>
</tr>
<tr>
<td>Cilium</td>
<td>auto</td>
<td>Fast</td>
<td>Very fast</td>
<td>Very fast</td>
</tr>
<tr>
<td>Flannel</td>
<td>auto</td>
<td>Very fast</td>
<td>Very fast</td>
<td>Very fast</td>
</tr>
<tr>
<td>Kube-OVN</td>
<td>auto</td>
<td>Fast</td>
<td>Very slow</td>
<td>Fast</td>
</tr>
<tr>
<td>Kube-router</td>
<td>none</td>
<td>Slow</td>
<td>Very slow</td>
<td>Slow</td>
</tr>
<tr>
<td>Weave Net</td>
<td>manual</td>
<td>Very fast</td>
<td>Very slow</td>
<td>Very fast</td>
</tr>
</tbody>
</table>

- **Configure MTU in Calico**
- **Use Wireguard for encryption. Minimal performance impact**
- **Leverage [benchmarking suite](#) for your own clusters**
Calico supports multiple data planes
Works with native cloud provider SDN/CNI
AirShip 2.0 Architecture

- Layer3 SDN for Control Plane, independent of the tenant network
- Integrate routing announcements for control plane directly into physical fabric
- Multi-NIC
- Optimized service load balancing
Next Generation Dataplane (eBPF)

- Full support for Calico network policies
- Networking optimization leveraging eBPF for traffic to/from Pods
- No kube-proxy needed (full support for kube proxy functionality e.g. node ports and services)
- Connection time load balancing for traffic originating from workloads
- Integrated with the Calico Policy engine
- Support for Direct Server Return optimization
  - DSR is to optimize the return path for traffic originating outside the cluster destined to workloads to avoid additional hops on the return path
  - Source IP preservation for public cloud
Architecture: Network Optimization

Node 1

Pod A

veth

tc BPF

iptables etc.

ct mop

(1) Contrack lookup

(2) Approved flow detected: bypass and set mark

(3) Program detects mark, skips processing

eth0

tc BPF

Node 2

Pod B

eth0

tc BPF

iptables etc.

tc BPF

veth

(4) Contrack lookup

(5) Approved flow detected: bypass and set mark

(6) Program detects mark, skips processing
Architecture: Policy Application
Architecture: Service Optimization
Calico/VPP integration

- VPP dataplane option for Calico
  - Transparent for users except for basic initial interface configuration
- Custom VPP plugins for K8s networking:
  - Optimized NAT plugin for service load balancing
  - Specific plugin for efficient Calico policies enforcement
- VPP configuration optimized for container environments:
  - Interrupt mode, SCHED_RR scheduling
  - No hugepages required
  - GRO / GSO support for container interfaces
Benefits

● Performance
  ○ World-class encryption performance: IPsec / Wireguard
  ○ Reduced overall CPU consumption

● Operational simplicity
  ○ Network stack decoupled from OS - easier to upgrade
  ○ VPP is packaged as a regular container
  ○ Very limited kernel dependencies

● Better control over resources dedicated to container networking

● Extensibility through VPP plugins
Software architecture

K8s API

Calico API

CNI Plugin

Felix
Policies rendering

Calico-VPP agent
CNI, BGP Routing, Services, Policies

VPP

Calico-node container

VPP container

Calico-VPP-node pod

: Regular Calico / K8s components

: VPP-specific components
References

- Calico: [https://www.projectcalico.org/](https://www.projectcalico.org/)
- Join the Calico/VPP slack to stay up to date!
  - [https://calicousers.slack.com/archives/C017220EXU1](https://calicousers.slack.com/archives/C017220EXU1)
- Check out our docs if you’d like to learn more or try it out:
  - [https://github.com/projectcalico/vpp-dataplane/wiki](https://github.com/projectcalico/vpp-dataplane/wiki)
- Calico dataplane driver for VPP:
  - Code: [https://github.com/projectcalico/vpp-dataplane](https://github.com/projectcalico/vpp-dataplane)
  - Slack channel: [https://calicousers.slack.com/archives/C017220EXU1](https://calicousers.slack.com/archives/C017220EXU1)
- 40Gbps pod-to-pod IPSec for Calico with VPP: