

Table of Contents

Foreword.....	xiii
Preface.....	xv
1. Why Cilium?.....	1
From Early CNIs to Modern Datapaths	2
Cilium: Origins and Evolution	3
Industry Adoption	4
Becoming a CNCF Graduated Project	4
An Open Source Success Story	4
Cilium's Use Cases	5
Networking with Cilium as a CNI	5
Ingress and Gateway API	6
Service Mesh Without the Sidecars	8
Multicluster Networking and Load Balancing	9
Interconnectivity with the Wider Network	9
Firewalling and Network Policy	10
Encryption and Secure Connectivity	11
Observability with Hubble	11
Looking Ahead	12
2. Inside Cilium.....	15
Cilium at a Glance	15
The Cilium Agent	18
The Cilium CNI Plugin	19
The Cilium Operator	22
eBPF Programs and eBPF Maps	22
The Cilium CLI	24

Hubble	25
The Envoy Proxy	26
The DNS Proxy	27
Summary	27
3. Getting Started with Cilium.....	29
Setting Up Your Environment	29
Kubernetes in Docker (kind)	30
Installing the Cilium CLI	30
Installing the Hubble CLI	30
Installing Helm	31
Deploying a kind Cluster	31
Installing Cilium	32
Installing Cilium Using the Cilium CLI	33
Installing Cilium Using Helm (Recommended)	34
Deploying a Sample Application	36
Securing the Application with Cilium Network Policies	41
Monitoring the Application with Hubble	44
Summary	48
4. IP Address Management.....	49
IPv4 and IPv6 Support in Cilium	49
IP Address Management	50
Pod IP Address Allocation	50
Kubernetes Host Scope IPAM Mode	52
Cluster Scope IPAM Mode	53
Multi-Pool IPAM Mode	55
ENI IPAM Mode	61
IPv6 Clusters	64
Dual Stack (IPv4/IPv6)	65
Single Stack (IPv6 Only)	68
Summary	71
5. The Cilium Datapath.....	73
Intranode Connectivity	75
Internode Connectivity	80
Native Routing Mode	80
Encapsulation Mode	85
Summary	92

6. Service Networking.....	93
Kubernetes Service Refresher	93
kube-proxy Refresher	94
kube-proxy Replacement	97
Agent Availability and Datapath Resiliency	102
KPR Service Behavior Support	104
Session Affinity	104
Traffic Policies	106
External-Facing Services	108
Summary	112
7. Ingress and Gateway API.....	113
Handling External Traffic in Kubernetes	114
Getting Started with Cilium Ingress	114
TLS Termination with Cilium Ingress	119
Ingress Limitations	121
Introducing Gateway API	122
Gateway API Objects	124
GatewayClass	124
Gateway	124
HTTPRoute	125
Routing HTTP Traffic with Gateway API	126
TLS Termination with Gateway API	129
Layer 7 Load Balancing Across Backends	130
50/50 Load Balancing	130
Adjusting Weights: 99/1 Split	132
Traffic Management	132
Modifying HTTP Headers with Gateway API	133
HTTP Redirects with Gateway API	135
gRPC Routing	138
East-West Layer 7 Routing with GAMMA and Gateway API	140
Summary	142
8. Performance Networking and Traffic Optimization.....	143
Service Traffic Distribution	145
Local Redirect Policy	149
Using LRP to Keep Traffic on the Same Node	150
Using LRP to Redirect DNS Traffic Locally	153
Direct Server Return	156
eXpress Data Path	160
Maglev Consistent Hashing	161

Netkit	167
Summary	169
9. Multicluster Networking.....	171
Multicluster Kubernetes	172
What Cluster Mesh Solves	173
Use Cases	173
Cluster Mesh Architecture	177
Application Datapath	178
Cluster Mesh API Server	180
Meshing Clusters	181
Preparing Clusters	181
Configuring Cilium for Cluster Mesh	182
Propagating TLS Trust	184
Testing Cluster Mesh	186
Global Services	187
Configuring Global Services	187
Endpoint Affinity	190
Global Policy	193
TLS Certificates in Cluster Mesh	196
Summary	199
10. Cluster Access.....	201
L2 Announcements	203
Setting Up the Environment	203
Configuring L2 Announcements	204
Applying the L2 Announcement Policy	205
Inspecting ARP Traffic	206
Simulating a Node Failure	209
BGP	211
Setting Up the Environment	212
Enabling Cilium BGP Control Plane	215
Applying the Policies and Verifying Route Advertisements	220
Summary	221
11. Cluster Egress.....	223
Masquerading	223
Configuring Masquerading	224
Masquerading Modes	226
Egress Gateway	229
Egress Gateway Use Cases	230

Cilium Egress Gateway Example Walk-Through	231
Bandwidth Manager	237
Summary	241
12. Network Policy	243
Creating a Basic Policy	243
Restricting Traffic by Port	244
Endpoints and Policy Enforcement	246
Default Deny	246
Egress Rules	247
Identity	248
The Foundations of Security	248
Identity in Cilium	249
The Policy Datapath	252
Selecting Identities with Policy	253
From Endpoints	253
Policy with Multiple Rules	254
Multiple Statements Versus Multiple Rules	256
Cluster-wide Policy	257
Reserved Labels, Reserved Identities, and Entities	258
Combining Statements	261
Beyond the Cluster	263
CIDR Ranges	263
CIDR Policy	263
Identity with Ingress and Gateway API	264
Deny Rules	265
Summary	266
13. Layer 7 and FQDN Policy	267
Layer 7 Policy	267
Securing HTTP	268
The Envoy Datapath	270
Performance	271
DNS Policy	274
FQDN Policy	276
Writing FQDN Policy	276
FQDN Identity	278
Content Delivery Networks	281
TLS Interception	282
Secrets and Security in Cilium	283
Writing HTTPS Policy	285

Public Trust	285
Internal CA	286
Overriding Pod CA Certificates	287
TLS Policy	289
Limitations	290
Summary	291
14. Transparent Encryption.....	293
Use Cases	293
Compliance	294
Threat Models	294
Traffic Flow	295
Encapsulation	297
Encryption and Authentication	299
Configuring and Validating Transparent Encryption	299
Testing Encryption	301
Viewing Unencrypted Traffic	301
Viewing Encrypted Traffic	302
Viewing Unencrypted Flows	303
Summary	304
15. Observability with Hubble.....	305
Hubble Architecture	305
Hubble Relay	307
Hubble CLI	308
Viewing Flow Details with JSON Output	311
Hubble Exporter	312
Layer 7 Visibility	313
Flow Redaction	317
Hubble UI	318
Exporting Metrics from Hubble	320
Summary	322
16. Operations.....	323
Installation and Lifecycle Management	323
Cilium CLI Versus Helm	324
Running Helm from Your Laptop or Bastion Host	324
Running Helm from Argo CD or Flux	325
Upgrades, Migrations, and Version Management	326
Upgrade or Replace?	327
Traffic Stability During Upgrade	327

Migration from Other CNIs	328
Monitoring, Capacity, and Observability	329
Day 2 Operations and Troubleshooting	333
Gateway API, Ingress, and Layer 7 Policy	335
L2 Announcements	336
Leader Election Timing	336
Client Traffic Redirection	337
BGP	337
Announcing Pod IPs Versus Service IPs	338
Reliability	338
Summary	339
Index.....	341

When we started working on Cilium, none of us imagined how far the journey would go. What began as an ambitious idea—to rethink networking and security in cloud native systems using eBPF—turned into a mission shared by an incredible group of engineers, users, and open source contributors. The journey of Cilium has been nothing short of amazing: from working with the very first wave of users trying to get Kubernetes into production, to partnering with some of the largest enterprises running massive fleets of Kubernetes clusters powered by Cilium and eBPF.

Cilium was built in the open, shaped by a community that believed in a better way of doing cloud native networking and security. Every design decision, feature, and improvement was influenced by real-world feedback from operators running production systems, contributors pushing boundaries, and users who trusted us with their most critical workloads. The community didn't just adopt Cilium—it helped build it. For that, I am deeply grateful.

I vividly remember the early days when simply getting a basic ping working between two Kubernetes pods felt like a breakthrough. The days when we had to support Docker, Kubernetes, and Mesos at the same time, because the ecosystem had not yet settled on a single orchestration standard. The excitement when we got traceflow working properly by implementing full ICMP support. The moment we scaled our first Kubernetes cluster to 5,000 nodes and realized we were entering an entirely new class of infrastructure. I remember the first Flubbe flow that worked and to end, unlocking a completely new way to observe and troubleshoot distributed systems. Donating Cilium to the CNCF and later celebrating its graduation marked a significant milestone that validated the technology and the strength and maturity of the community behind it.

And I remember the moment when we realized Cilium was no longer just an exciting open source project. Today, Cilium is critical infrastructure powering stock exchanges, massive AI training clusters, sports stadiums, high scale gaming platforms, financial institutions, and some of the most demanding production environments in the world. We do not take that responsibility lightly.