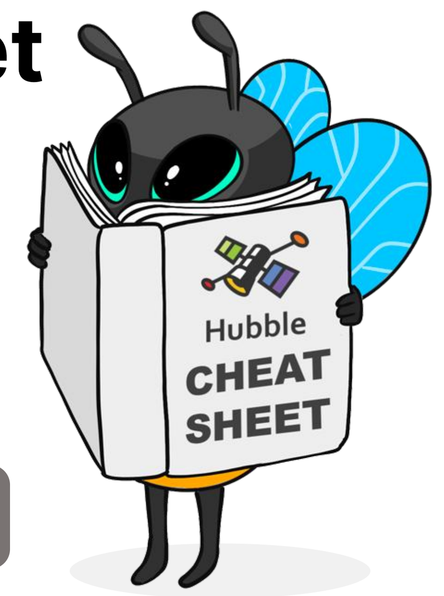
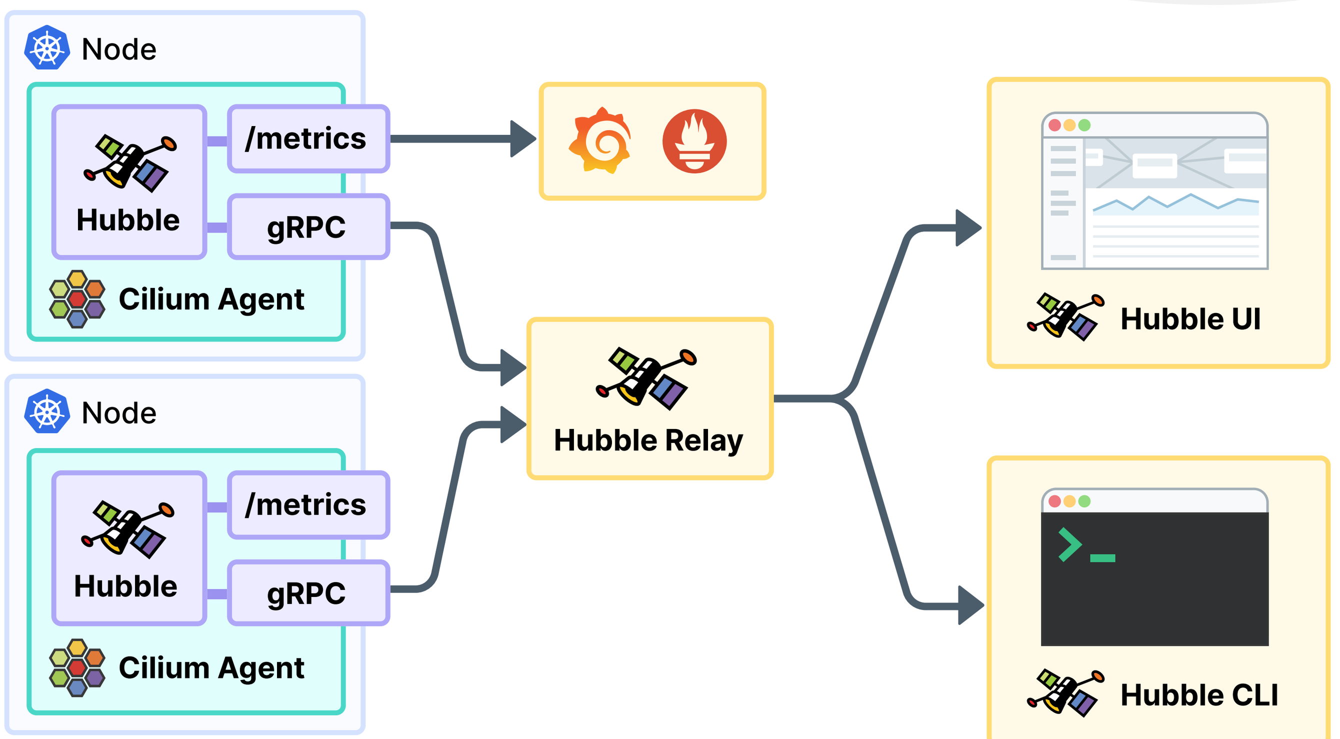
Hubble Cheat Sheet



Hubble can provide information on the following areas:

- Service dependencies and communication maps
- Network Monitoring and alerting
- Application Monitoring
- Security Observability

Components of Hubble



Cilium Agent - Runs the cilium-agent binary which acts as a CNI to manage connectivity, observability, and security for all CNI-managed Kubernetes pods.
Hubble Relay - Provides a cluster-wide API for querying Hubble flow data, which can be accessed directly or via the Hubble CLI and UI.
Hubble UI - Provides a graphical UI for visualizing network flow data, network policy, and security-related events.

Accessing Hubble

To access the CLI

```
$ cilium hubble port-forward&
```

```
Forwarding from 0.0.0.0:4245 -> 4245
Forwarding from [::]:4245 -> 4245
```

To access the UI

```
$ cilium hubble ui
```

```
Forwarding from 0.0.0.0:12000 -> 8081
Forwarding from [::]:12000 -> 8081
```

Checking Hubble Status

Seeing the current and max flows at 100% is expected, as the Hubble relay ring buffer fills, older events will automatically be dropped.

```
$ hubble status
```

```
Healthcheck (via localhost:4245): Ok
Current/Max Flows: 11917/12288 (96.98%)
Flows/s: 11.74
Connected Nodes: 3/3
```

ISOVALENT
now part of CISCO

Creators of eBPF cilium tetragon

Selecting which traffic flows to observe from the buffer

```
$ hubble observe
```

--all	Get all flows stored in Hubble's buffer
--first N	Get first N flows stored in Hubble's buffer
-f, --follow	Follow flows output
--last N	Get last N flows stored in Hubble's buffer (default 20)

OBSERVING AND FILTERING TRAFFIC EXAMPLES

Observe by Resource

With `$ hubble observe` you can filter by resources, either looking at incoming/outgoing or all traffic for that resource, below is a list of the filters available.

--from-label filter	Show only flows originating in an endpoint with the given labels (e.g. "key1=value1")
--from-namespace filter	Show all flows originating in the given Kubernetes namespace
--from-pod filter	Show all flows originating in the given pod name prefix([namespace/]<pod-name>). If namespace is not provided, 'default' is used
--from-port filter	Show only flows with the given source port (e.g. 8080)
--from-service filter	Show flows where the source IP address matches the ClusterIP address of the given service name prefix([namespace/]<svc-name>).
-l, --label filter	Show only flows related to an endpoint with the given labels (e.g. "key1=value1")
-n, --namespace filter	Show all flows related to the given Kubernetes namespace
--node-name filter	Show all flows which match the given node names (e.g. "k8s*", "test-cluster/*.company.com")
--not filter[=true]	Reverses the next filter to be blacklist i.e. --not --from-ip 2.2.2.2
--pod filter	Show all flows related to the given pod name prefix ([namespace/]<pod-name>). If namespace is not provided, 'default' is used.
--service filter	Show flows where either the source or destination IP address matches the ClusterIP address of the given service name prefix ([namespace/]<svc-name>)
--to-label filter	Show only flows terminating in an endpoint with given labels (e.g. "key1=value1")
--to-namespace filter	Show all flows terminating in the given Kubernetes namespace.
--to-pod filter	Show all flows terminating in the given pod name prefix([namespace/]<pod-name>). If namespace is not provided, 'default' is used
--to-port filter	Show only flows with the given destination port (e.g. 8080)
--to-service filter	Show flows where the destination IP address matches the ClusterIP address of the given service name prefix ([namespace/]<svc-name>)

The following examples will show a mix of these filters in use

Observe by Protocol

--protocol filter	Show only flows which match the given L4/L7 flow protocol (e.g. "udp", "http")

```
$ hubble observe --pod deathstar --protocol http
```

```
May 4 13:23:40.501: default/tiefighter:42690 ->
default/deathstar-c74d84667-cx5kp:80 http-request FORWARDED
(HTTP/1.1 POST
http://deathstar.default.svc.cluster.local/v1/request-landing)
```

```
$ hubble observe --namespace tenant-jobs --protocol dns
```

```
Aug  3 15:13:18.943: tenant-jobs/coreapi-767cf69fb8-cvqxl:53740
(ID:44253) -> kube-system/coredns-787d4945fb-6vvfg:53 (ID:43153)
dns-request proxy FORWARDED (DNS Query
elasticsearch-master.tenant-jobs.svc.cluster.internal. AAAA)
```

Observe by Policy Verdict

--verdict filter	Show only flows with this verdict [FORWARDED, DROPPED, AUDIT, REDIRECTED, ERROR, TRACED, TRANSLATED]

```
$ hubble observe --pod deathstar --verdict DROPPED
```

```
May 4 13:23:47.852: default/xwing:42818 <>
default/deathstar-c74d84667-cx5kp:80 Policy denied DROPPED
(TCP Flags: SYN)
```

Observe by FQDN

--fqdn filter	Show all flows related to the given fully qualified domain name (e.g. "*.cilium.io").
--from-fqdn filter	Show all flows originating at the given fully qualified domain name (e.g. "*.ebpf.io").
--to-fqdn filter	Show all flows terminating at the given fully qualified domain name (e.g. "*.isovalent.com").

```
$ hubble observe --to-fqdn api.github.com
```

```
Aug 3 15:12:13.929: tenant-jobs/crawler-5c645d68f4-qchk8:47180
(ID:21067) -> api.github.com:80 (world) policy-verdict:all
EGRESS ALLOWED (TCP Flags: SYN)
```

Observe by HTTP Method, Path and Status

--http-method filter	Show only flows which match this HTTP method (e.g. "GET", "POST")
--http-path filter	Show only flows which match this HTTP path regular expressions (e.g. "/page/\\d+")
--http-status filter	Show only flows which match this HTTP status code prefix (e.g. "404", "5+")

```
$ hubble observe --namespace tenant-jobs --http-path
/applicants
```

```
Aug 3 15:16:18.351: tenant-jobs/resumes-86bbf46b88-n6mcn:51768
(ID:20808) -> tenant-jobs/coreapi-767cf69fb8-cvqxl:9080
(ID:44253) http-request FORWARDED (HTTP/1.1 POST
http://coreapi:9080/applicants)
```

```
$ hubble observe --label app=resumes --http-method POST
```

```
Aug 3 15:16:28.591: tenant-jobs/resumes-86bbf46b88-n6mcn:51768
(ID:20808) <- tenant-jobs/coreapi-767cf69fb8-cvqxl:9080
(ID:44253) http-response FORWARDED (HTTP/1.1 200 33ms (POST
http://coreapi:9080/applicants))
```

A more complex example

Filters can be combined, too, the below example filters for flows of **HTTP** requests any pod with the label `"app=core-api"`, where the **HTTP** path is `"/applicants"` and the **HTTP** method is `"PUT"`

```
$ hubble observe --namespace tenant-jobs  --from-label
'app=coreapi' --protocol http --http-path /applicants
--http-method PUT
```

```
Aug 3 15:26:41.563: tenant-jobs/coreapi-767cf69fb8-cvqxl:49662
(ID:44253) -> tenant-jobs/elasticsearch-master-0:9200 (ID:16821)
http-request FORWARDED (HTTP/1.1 PUT
http://elasticsearch-master.tenant-jobs.svc.cluster.local:9200/
applicants/_create/827)
```

You can use the following argument to exclude data from results:

--not filter[=true]	Reverses the next filter to be blacklist i.e. --not --from-ip 2.2.2.2

This example command ensures no flows from anything with a specific label are returned when viewing all flows from a namespace

```
$ hubble observe -n tenant-jobs --not --label app=coreapi
```

Formatting the output

--color string	Colorize the output when the output format is one of 'compact' or 'dict'. The value is one of 'auto' (default), 'always' or 'never' (default "auto")
-o, --output string	Specify the output format, one of: **compact:** Compact output **dict:** Each flow is shown as KEY:VALUE pair **jsonpb:** JSON encoded GetFlowResponse according to proto3's JSON mapping **json:** Alias for jsonpb **table:** Tab-aligned columns (default "compact")
--color string	Colorize the output when the output format is one of 'compact' or 'dict'. The value is one of 'auto' (default), 'always' or 'never' (default "auto")
--ip-translation	Translate IP addresses to logical names such as pod name, FQDN, ... (default true)

Get hands on with Cilium, Hubble and eBPF at isovalent.com/labs