

Kris Buytaert @krisbuytaert

Bow for me, for I am coroot

OSMC 2024

Who am I ?

- I used to be a developer
- Then I became an Ops person
- Chief Trolling/Travel/Technical Officer @ Inuits.eu
- Chief Yak Shaver @ o11y.eu
- Organiser of #devopsdays, #cfgmgmtcamp, #loadays, ...
- Cofounder of all of the above
- Everything is a Freaking DNS Problem
- DNS : devops needs sushi
- @krisbuytaert on twitter/github/
- @mastodon.social/@krisbuytaert

Why this title ?



Observability is the new Hype

- Docker, Docker, Docker, Docker, Docker, Docker...
- Kubernetes, Kubernetes, Kubernetes, Kubernetes,
Kubernetes, Kubernetes,
- O11y, o11y, o11y, o11y, o11y, o11y, o11y,
- We're all doing this, right ?
- This is the new default, right . ?

What is monitoring?

- High level overview of the state of a service/component
- Availability
- Technical components
- Performance ?

What is going on?

What's observability?

- Understand how your services behave
- Like you are at their place
- Without incident specific code

Why is this going on?

What's observability - in Practice?

Are there Three pillars:

- Metrics
- Logs
- Traces

The Fourth Pillar



Maybe it's not Pillars, but layers.

How do monitoring and observability connect?

- Monitoring is required
- If lucky, monitoring is enough
- Observability is removing luck <- @roidelapluie
- It's the next layer up ...

Remember Packetbeat ?

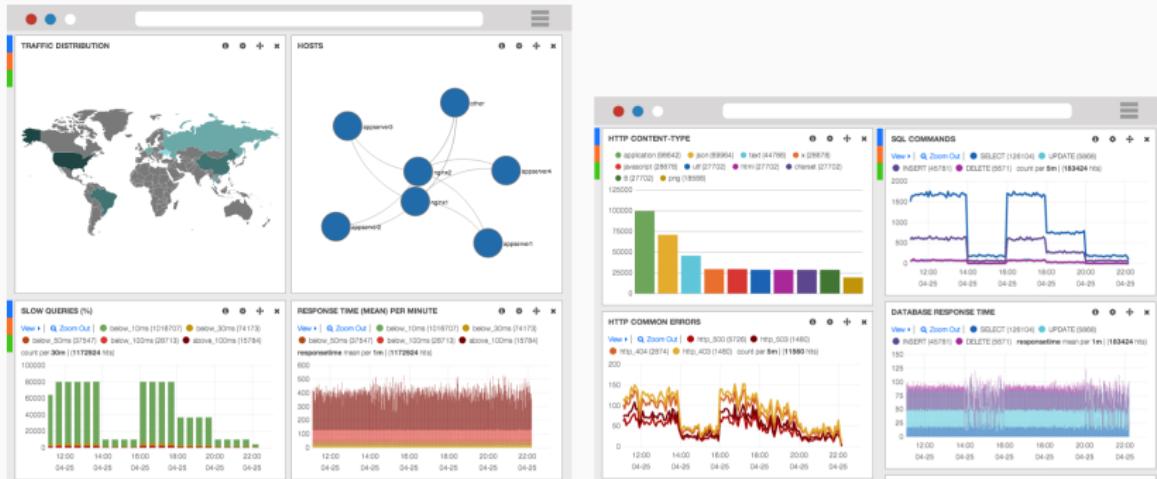


Figure 2: Nostalgia

Where is the real observability ?

- Often we have metrics
- But only for a month
- Often we lost our long term metrics
- Often we have logs
- But no derived metrics
- We are only alerting on those metrics
- We are not learning from our metrics
- We've regressed

Please Fix your Monitoring first ...

- SSOT , 100% automated monitoring
- No manual configuration tolerated
- Fix your metrics
- I bet you have regression on shipping your Metrics
- Fix your log shipping
- I bet you logshipping is partially broken
- I bet you have broken dashboards

- Protocol
- SDK's
- Signals
 - Traces (stable)
 - Metrics (stable)
 - Logs (stable)
 - Profiles (wip Q1 2025)
 - RUM (wip)
- OtelCollector



- Top CNCF Project
- “Vendor Driven Standardisation”
- Protocol, not implementation
- Claims AutoInstrumentation,
- Different components have different maturity levels
- Different languages have different maturity levels (11)

There is no magic



Paul Johnston - current focus: moving electrons
@PaulDJohnston

...

This.

Cannot believe how many people think that generalised instrumentation is somehow the "best" solution.

"What do I need to know? It would be good to instrument exactly that wouldn't it!"

Getting data and querying is hard.

Producing the data you need in the first place?



Luc van Donkersgoed @donkersgood · Sep 10

Replying to @dfrasca80 and @honeycombio

Not a silly question at all! OpenTelemetry supports auto instrumentation, but I'm a strong proponent for manual instrumentation. This allows me to add business context which makes the output WAY more valuable.

Is eBPF the magic ?

- Linux Kernel level
- extended Berkeley Packet filter
- “A marriage of strace + tcpdump on steroids”
- Framework to attach to low level functions
- Framework used to collect statistics
- Framework to build a monitoring tool



Open Source ebpf Tools

- ebpf_exporter
- opentelemetry-network
- Falco (security)
- Coroot
- Pixie (k8s)
- Odigos (k8s)
- Tracee (docker/k8s)
- Caretta (maps your dependencies) (k8s)
- Keppler (power usage / k8s)

The eBPF tooling landscape

- K8s only ..
- Reality is mixed ..

So most of the tools are not useful for the majority of our industry



- Open source (Apache 2.0)
- Cloud-agnostic : bare metal, k8s ...
- eBPF based exporter
- Ships : metrics, logs and traces
- zero instrumentation
- application AND eBPF based tracing
- SLO Driven alerting
- Fast insight in problems
- Root Cause Detection

Coroot requirements

- Prom Compatible datastore for Metrics
 - e.g Prometheus, Thanos, Mimir, Cortex, Victoria ...
 - per project
 - label value too long for metric (actual: 3741, limit: 2048) , configure it large enough
 - used to store metrics
- Clickhouse
 - schema per project
 - Recently added support for clustered tables (still need to test)
 - used to store logs, traces, ...
- pgsql (for prod)
 - used to store project information



Prometheus

- Prometheus is an Open Source CNCF Project
- Collects and stores metrics - Pull-based
- Framework , not integrated solution
- Exporters for every piece of the infra
- Long-Term Support release (powered by o11y.eu)

Prometheus Compatible MultiTenant Backends

- Thanos



- Cortex



- Mimir



- ...

Clickhouse

“ClickHouse is an open-source column-oriented DBMS (columnar database management system) for online analytical processing (OLAP) that allows users to generate analytical reports using SQL queries in real-time.”

- Apache 2.0
- SQL all the things



Clickhouse & Otel

```
clickhouse01.ollylab.olly.eu :) show tables;
```

```
SHOW TABLES
```

```
Query id: 6376d37a-a6b4-4052-92ab-dc1d66713e03
```

	name
1.	otel_logs
2.	otel_traces
3.	otel_traces_trace_id_ts
4.	otel_traces_trace_id_ts_mv
5.	profiling_profiles
6.	profiling_profiles_mv
7.	profiling_samples
8.	profiling_stacks

```
8 rows in set. Elapsed: 0.003 sec.
```

Clickhouse TTL

```
clickhouse01.olillylab.olilly.eu :) show create table otel_logs;

SHOW CREATE TABLE otel_logs

Query id: 0badcedc-776f-4d06-b0e7-4f9b8a6e4685

statement

1. | CREATE TABLE coroot_olillylab.otel_logs
(
    `Timestamp` DateTime64(9) CODEC(Delta(8), ZSTD(1)),
    `TraceId` String CODEC(ZSTD(1)),
    `SpanId` String CODEC(ZSTD(1)),
    `TraceFlags` UInt32 CODEC(ZSTD(1)),
    `SeverityText` LowCardinality(String) CODEC(ZSTD(1)),
    `SeverityNumber` Int32 CODEC(ZSTD(1)),
    `ServiceName` LowCardinality(String) CODEC(ZSTD(1)),
    `Body` String CODEC(ZSTD(1)),
    `ResourceAttributes` Map(LowCardinality(String), String) CODEC(ZSTD(1)),
    `LogAttributes` Map(LowCardinality(String), String) CODEC(ZSTD(1)),
    INDEX idx_trace_id TraceId TYPE bloom_filter(0.001) GRANULARITY 1,
    INDEX idx_res_attr_key mapKeys(ResourceAttributes) TYPE bloom_filter(0.01) GRANULARITY 1,
    INDEX idx_res_attr_value mapValues(ResourceAttributes) TYPE bloom_filter(0.01) GRANULARITY 1,
    INDEX idx_log_attr_key mapKeys(LogAttributes) TYPE bloom_filter(0.01) GRANULARITY 1,
    INDEX idx_log_attr_value mapValues(LogAttributes) TYPE bloom_filter(0.01) GRANULARITY 1,
    INDEX idx_body Body TYPE tokenbf_v1(32768, 3, 0) GRANULARITY 1
)
ENGINE = MergeTree
PARTITION BY toDate(Timestamp)
ORDER BY (ServiceName, SeverityText, toUnixTimestamp(Timestamp), TraceId)
TTL toDate(Timestamp) + toIntervalDay(1)
SETTINGS index_granularity = 8192, ttl_only_drop_parts = 1 |
```

Coroot deployment options

- helm (k8s)
- puppet
- docker
- docker swarm
- (curl scripts)
- (manually)
- (ping me if you want rpms)

Deploying Coroot

- Install the server,
- Initial connection for default project to prometheus / clickhouse
- Create more individual projects,
- Configure their prometheus / clickhouse backends

Coroot-node-agent

- Install the agent
- configure it to connect to the server with the APIkey for your project
- ships ebpf based metrics / logs / traces to the server
- is an actual prometheus exporter (:10300/metrics)

Additional support for instrumentation of

- MySQL
- PgSQL
- Redis
- Mongo

requires coroot-cluster-agent (per project)

Database Integration Use case

- More in depth details about database
- (more on this incident later)

The screenshot shows the OpenTelemetry (OTel) UI interface. At the top, there's a navigation bar with a green header containing the text "coroot:~#". To the right of the header are several icons: a dropdown for "esodev", a search bar with placeholder text "search for apps and nodes", and other standard UI controls like a question mark icon, a clock icon labeled "last hour", a gear icon, and a user profile icon.

Below the header, there are tabs for "HEALTH", "INCIDENTS", "SERVICE MAP", "TRACES", "NODES", "DEPLOYMENTS", and "COSTS". The "INCIDENTS" tab is currently selected and underlined.

Under the "INCIDENTS" tab, there are search and filter controls. On the left is a search bar with placeholder "search" and a dropdown menu labeled "namespaces". On the right are two checked checkboxes: "application" and "monitoring", followed by a plus sign for more filters.

Below these controls, there are filtering options: "Critical", "Warning", "Resolved", and a checkbox for "Show resolved".

The main content area displays a table of incidents. The columns are: Incident, Application, Opened at (sorted by duration), Duration, Availability, Latency, Affected requests, and Consumed error budget. One incident is listed:

Incident	Application	Opened at ↓	Duration	Availability	Latency	Affected requests	Consumed error budget
i-kiptiauI	mysqld	Jul 19, 15:33:26 (58d ago)	58d	58.28%	100%	42 %	4172 %

At the bottom of the table, there are pagination controls: "Rows per page: 50", "1-1 of 1", and navigation arrows.

Projects & Namespaces

- projects map to agents with api-key
- isolated views
- namespaces for k8s / nomad

The screenshot shows the otly interface for the 'esodev' project. At the top, there's a search bar and tabs for 'HEALTH' (selected), 'INCIDENTS', 'TRACES', 'NODES', and 'DEPLOYMENTS'. Below the tabs, a search bar shows 'o11y'. A dropdown menu is open over the 'o11y' entry, listing 'nomaddev', 'nomadlab', and 'o11y'. The 'o11y' entry is highlighted. Below the dropdown, there's a summary section with metrics: 0 SLO violation, 3 Warning, and 15 OK. Underneath, a table lists applications: auditbeat, jaeger-agent, and postfix. The auditbeat row shows 'golang' as the language, 100% errors, 51ms latency, 2/2 instances, and 2 restarts.

coroot :~# nomaddev ~

This screenshot shows the otly interface for the 'nomaddev' project. The top navigation and search bar are identical to the previous screen. The 'HEALTH' tab is selected. A dropdown menu is open over the 'namespaces' button, listing 'empty (26)', 'external (2)', 'multis-infra (39)', and 'tomlfile (1)'. The 'empty (26)' option is selected. Below the dropdown, a table lists applications: localhost:3306, localhost:8500, mysql, auditbeat, and backup-nomad-jobs. The auditbeat row shows 'golang' as the language.

General Environment Health

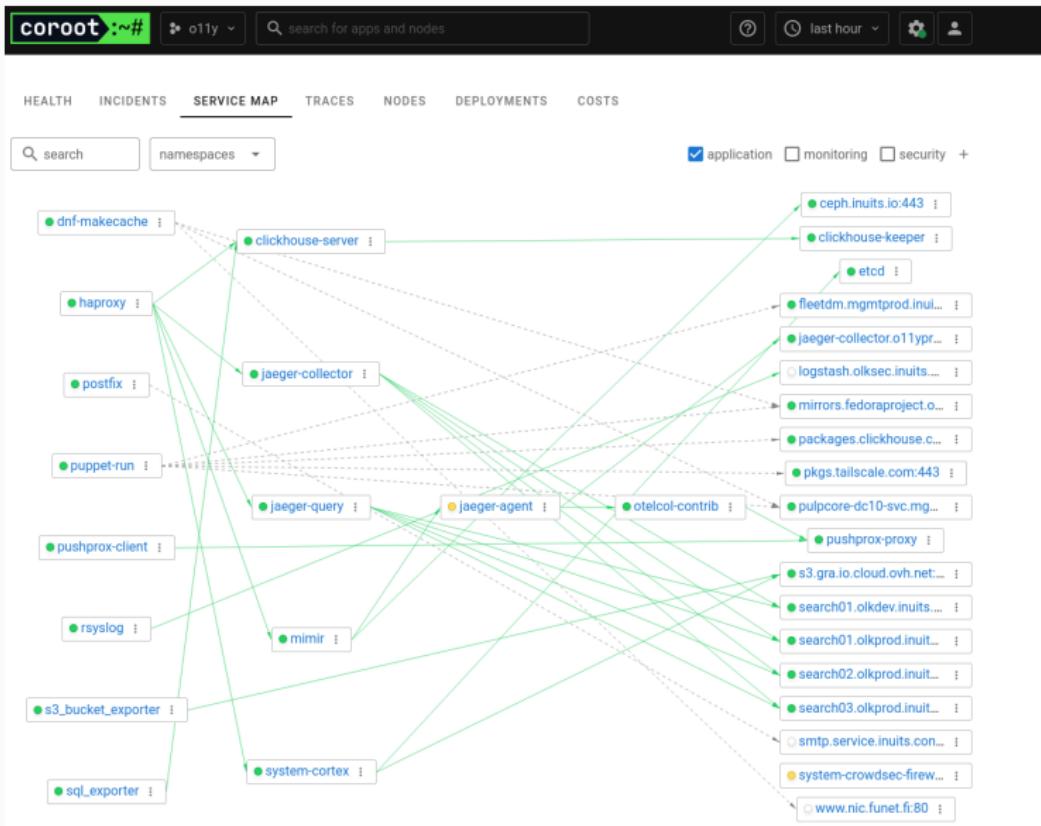
HEALTH SERVICE MAP TRACES NODES DEPLOYMENTS

search namespaces ▾ application monitoring +

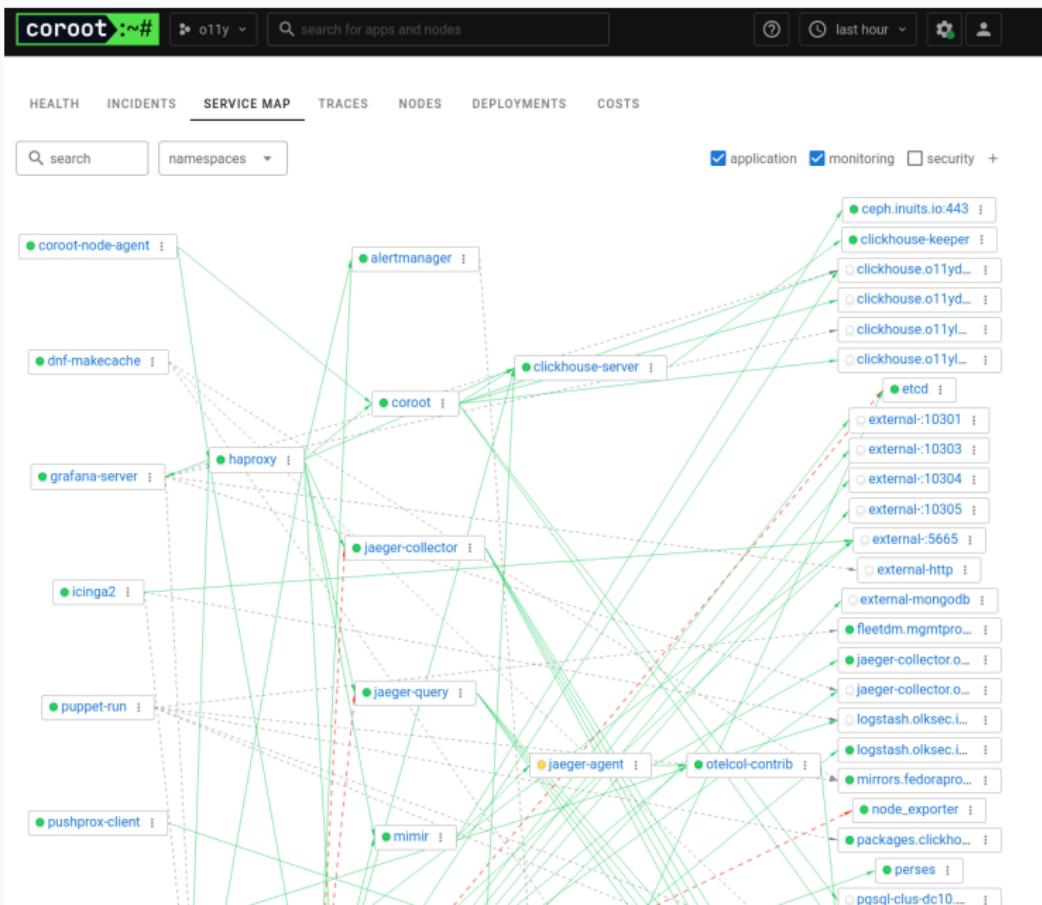
5 SLO violation 53 Warning 5 Errors in logs 44 OK

Application	Errors	Latency	Upstreams	Instances	Restarts	CPU	Mem	I/O	Disk	Net	L
consul	-	10s	55/55	6/6	-	-	-	2%	90%	25ms	
mysql	12%	25ms	-	3/3	-	-	-	<1%	82%	10ms	
mysql-06ebec06-c4ff-902f-80e6-36a...	13%	192ms	0/1	1/1	-	-	-	<1%	3%	failed connns	
mysql-45f329ab-2ef3-29d1-5538-cc...	9%	74ms	0/2	1/1	-	-	-	<1%	3%	25ms	
mysql-4fcfd407-cf7e-8810-a205-05f...	8%	92ms	0/1	1/1	-	-	-	<1%	2%	failed connns	
auditbeat	-	-	-	7/7	-	-	-	2%	93%	26ms	
backup-nomad-jobs-5b60a0b5-3b73...	-	-	1/1	0/1	-	-	-	-	-	-	
compactor-160cd7ab-9630-3164-ef...	-	5ms	0/1	1/1	-	-	-	-	-	failed connns	
compactor-6491125d-0e18-7c9e-70...	-	5ms	-	1/1	-	-	-	-	-	failed connns	
crowdsec-firewall-bouncer	-	-	1/1	10/10	2158	-	-	<1%	82%	44ms	
distributor-048a9392-dbcd-de17-db...	-	49ms	3/4	1/1	-	-	-	-	-	failed connns	
distributor-62235234-e70b-f793-a09...	-	49ms	1/2	1/1	-	-	-	-	-	failed connns	
distributor-772c1fa9-10e2-1d37-b22...	<1%	49ms	3/4	1/1	-	-	-	-	-	failed connns	
distributor-7a93fd91-f783-d893-b60...	<1%	50ms	3/4	1/1	-	-	-	-	-	failed connns	
distributor-daf5993e-f588-a13a-580...	-	49ms	3/4	1/1	-	-	-	-	-	failed connns	
distributor-e26a2c2d-bf29-6c41-95b...	<1%	49ms	3/4	1/1	-	-	-	-	-	failed connns	
dnf-makecache	-	-	1/1	0/11	-	-	-	-	-	-	
icinga2	-	-	3/4	17/17	-	-	-	2%	93%	failed connns	
ingester-559935ea-58f0-4786-f529-...	-	5ms	0/1	1/1	-	-	-	<1%	2%	failed connns	
innetester-ac391338-00fa-f51b-a3bc...	-	50ms	0/1	1/1	-	-	-	<1%	3%	failed connns	

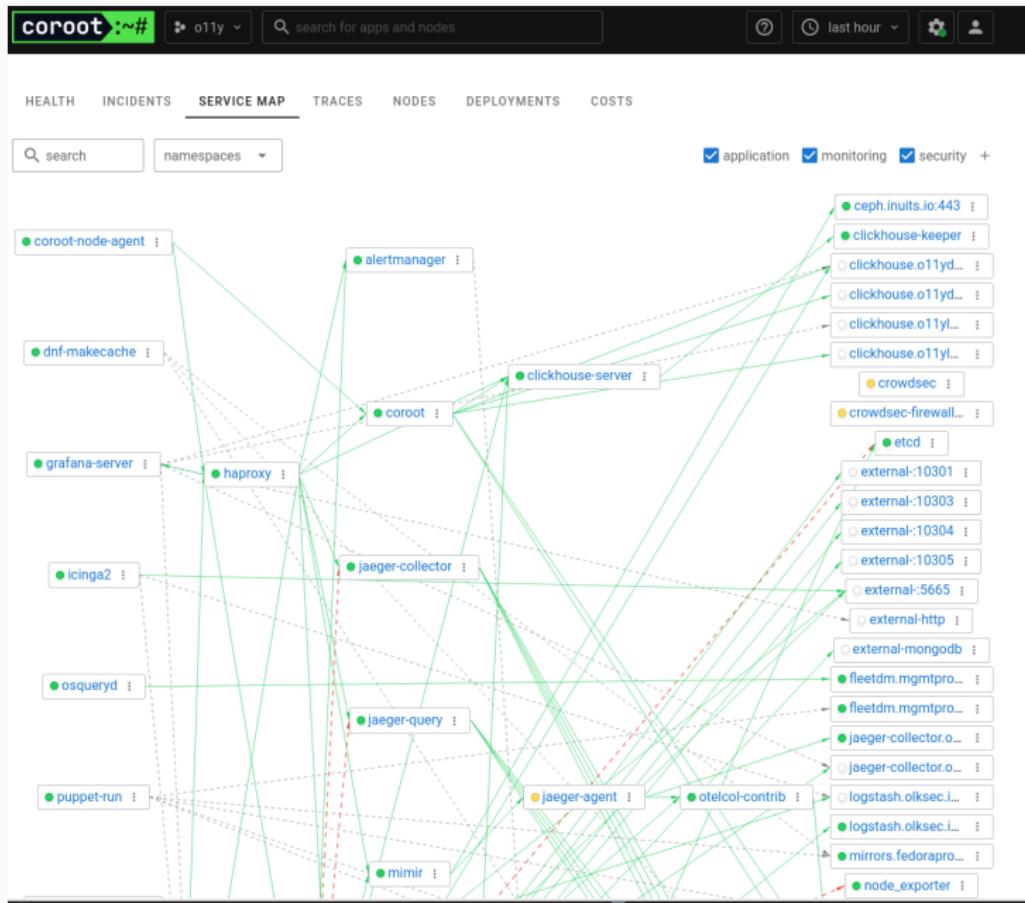
Service Map (Application)



Service Map (App + Monitoring)



Service Map (App + Monitoring + Security)



Filter and Define Categories



Configuration

GENERAL PROMETHEUS CLICKHOUSE AWS INSPECTIONS **APPLICATIONS** NOTIFICATIONS ORGANIZATION

Application categories ⓘ

You can organize your applications into groups by defining [glob patterns](#) in the <namespace>/<application_name> format.

Category	Patterns	Notify of deployments	Actions
application	The default category containing applications that don't fit into other categories	on	
control-plane	<code>kube-system/*</code> <code>*/kubelet</code> <code>*/kube-apiserver</code> <code>*/k3s</code> <code>*/k3s-agent</code> <code>*/systemd*</code> <code>*/containerd</code> <code>*/docker*</code> <code>**chaos*</code> <code>istio-system/*</code> <code>amazon-cloudwatch/*</code> <code>karpenter/*</code> <code>cert-manager/*</code> <code>argocd/*</code> <code>flux-system/*</code> <code>linkerd/*</code> <code>vault/*</code> <code>keda/*</code> <code>keycloak/*</code> <code>longhorn-system/*</code> <code>calico-system/*</code> <code>esm-cache</code> <code>/*motd*</code> <code>/*apt*</code> <code>/*fwupd*</code> <code>/*snap*</code>	off	
monitoring	<code>monitoring/*</code> <code>prometheus/*</code> <code>/*prometheus*</code> <code>grafana/*</code> <code>/*grafana*</code> <code>/*alertmanager*</code> <code>coroot/*</code> <code>/*coroot*</code> <code>metrics-server/*</code> <code>/*signalilo*</code> <code>/*icinga*</code>	off	
cluster-agent	<code>/*coroot-cluster-agent*</code>	off	
security	<code>/*crowdsec*</code> <code>/*osqueryd*</code>	off	

[Add a category](#)

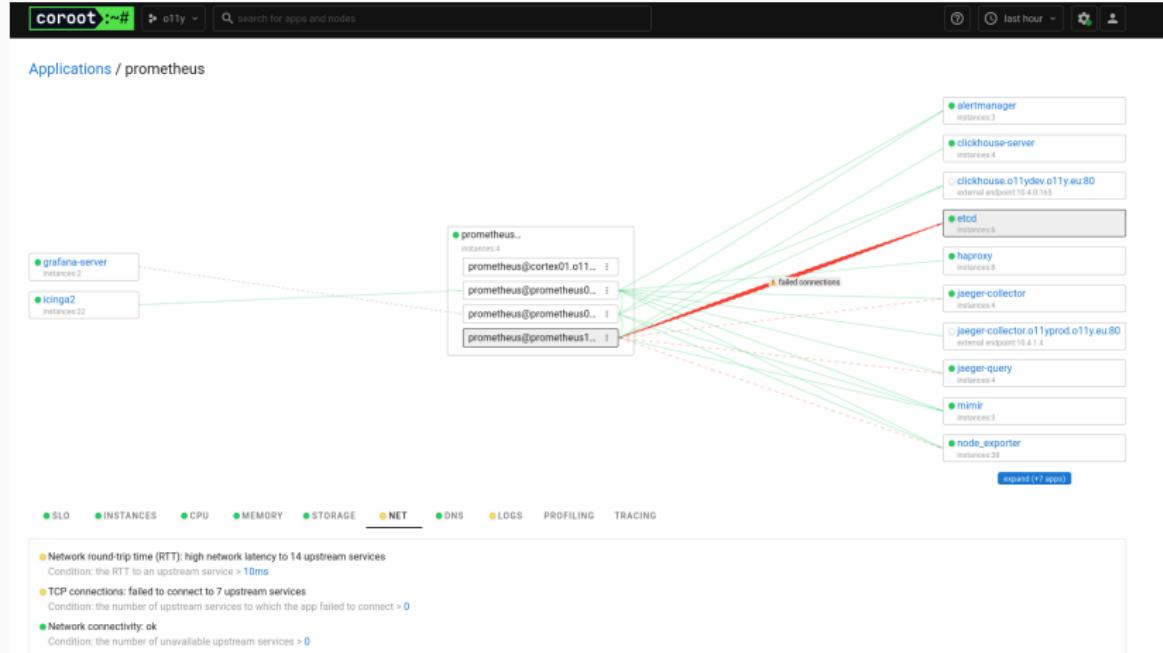
Custom applications

Coroot groups individual containers into applications using the following approach:

- **Kubernetes metadata:** Pods are grouped into Deployments, StatefulSets, etc.
- **Non-Kubernetes containers:** Containers such as Docker containers or Systemd units are grouped into applications by their names. For example, Systemd services named `mysql` on different hosts are grouped into a single application called `mysql`.

This default approach works well in most cases. However, since no one knows your system better than you do, Coroot allows you to manually adjust application groupings to better fit your specific needs. You can match desired application instances by defining [glob patterns](#) for `instance_name`. Note that this is not applicable

Service Map Etcdb Error



DNS

coreot:~# ⌂ o11y ⌂ search for apps and nodes

last hour | 7 instances | expanded (+7 apps)

SLO INSTANCES CPU MEMORY STORAGE NET DNS LOGS PROFILING TRACING

● DNS latency: ok
Condition: the 95th percentile of DNS response times > 100ms

● DNS server errors: ok
Condition: the number of server DNS errors (excluding NXDOMAIN) > 0

● DNS NXDOMAIN errors: ok
Condition: the number of the NXDOMAIN DNS errors (for previously valid requests) > 0

Domain	Requests	No results (IPv4: A)	No results (IPv6: AAAA)	No results (other)	Send fail
cortex01.o11ylab.o11y.eu.o11ylab.o11y.eu	1.5 s	753	753	—	—
cortex03.o11ylab.o11y.eu.o11ylab.o11y.eu	1.4 s	711	710	—	—
cortex02.o11ylab.o11y.eu.o11ylab.o11y.eu	1.4 s	700	700	—	—
puppet5-db.mgmprod.inuits.eu	1.1 s	—	—	—	—
cortex03.o11ydev.o11y.eu	888	—	—	—	—
o11ylab-cortex.o11ydev.o11y.eu	812	—	—	—	—
cortex01.o11ylab.o11y.eu	753	377	376	—	—
cortex01.o11ydev.o11y.eu	751	—	—	—	—
cortex03.o11ylab.o11y.eu	710	355	355	—	—
cortex02.o11ylab.o11y.eu	700	350	350	—	—

DNS requests by type, per second

TypeA (blue), TypeAAAA (orange), TypePTR (teal)

DNS errors, per second

TypeArandomize (blue), TypeAAAMxdomain (orange)

DNS latency, seconds

Incidents

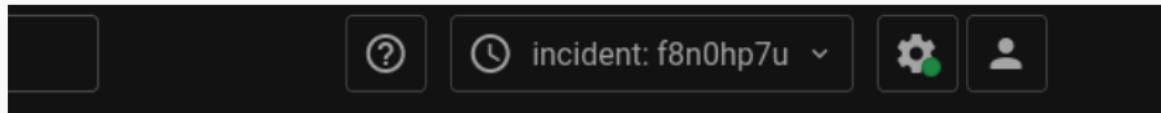
- All in one space ..

The screenshot shows the Coroot application interface. At the top, there's a header bar with the title 'coroot' and a search bar labeled 'search for apps and nodes'. Below the header are tabs for 'HEALTH', 'INCIDENTS' (which is selected), 'SERVICE MAP', 'TRACES', 'NODES', 'DEPLOYMENTS', and 'COSTS'. The main area displays a table of incidents:

Incident	Application	Opened at	Duration	Availability	Latency	Affected requests	Consumed error budget
1-z3i9zpdh	pulpcore-dc10-svc.mgmtprod.rnults.eu:8142	Nov 17, 02:34:37 (9h8m ago)	9h5m	100%	90.74%	9.3 %	926 %
1-t5egz2gn	nomad	Oct 22, 16:51:48 (25d ago)	25d			0.00 %	0.00 %
1-1jvtj0v	localhost:8500	Sep 13, 13:21:13 (64d ago)	64d	100%	66.8%	33 %	3320 %
1-x9cw71b8	localhost:3306	Sep 13, 13:21:12 (64d ago)	64d	58.18%	100%	42 %	4162 %
1-67phjy91	mysql	Sep 09, 10:40:33 (9d ago)	69d	91.55%	100%	8.5 %	845 %

At the bottom right of the table, there are buttons for 'Rows per page' (set to 50), '1-5 of 5', and navigation arrows. To the right of the table, there are filter buttons for 'application' and '+'.

- Incident timeframe.



MySQL Incident

coroot:~# nomaddev search for apps and nodes incident: z3i9zpdh ? 🕒

Applications / localhost:3306

mysqld_exporter instances:3

localhost:3306... external endpoint:127.0.0.1
127.0.0.1:3306 :

SLO INSTANCES CPU MEMORY MYSQL LOGS PROFILING TRACING

- Availability: error budget burn rate is 41.7x within 1 hour
Condition: the successful request percentage < 99%
- Latency: error budget burn rate is 0.0x within 1 hour
Condition: the percentage of requests served faster than 500ms < 99%

Client	Requests	Latency	Errors
mysqld_exporter mysql	0.2 /s	0.4 ms	0.07 /s

Latency & Errors heatmap, requests per second

errors >10s 10s 5s 2.5s 1s 0.5s 0.25s 100ms 50ms 25ms 10ms 5ms 0

0.004/s 0.1/s 0.2/s

Nov 17 02:00 03:00 04:00 05:00 06:00 07:00 08:00 09:00 10:00 11:00

OTIy 40

Tracing

- ebpf
- otel endpoint

MySQL Traces

coroot :~# esdev search for apps and nodes ? incident: k1ptiau1 🔍

OpenTelemetry (eBPF)

Latency & Errors heatmap, requests per second

errors
>10s
10s
5s
2.5s
1s
0.5s
0.25s
100ms
50ms
25ms
10ms
5ms
0

Jul 20 Jul 25 Jul 30 Aug 04 Aug 09 Aug 14 Aug 19 Aug 24 Aug 29 Sep 03 Sep 08 Sep 13

Trace ID	Client	Status	Duration	Name	Details
9fb4877e	mysqld_exporter	🔴 ERROR	0.0 ms	query	SHOW ALL SLAVES STATUS
af0e7c60	mysqld_exporter	🔴 ERROR	0.0 ms	query	SHOW ALL SLAVES STATUS NOLOCK
f7362ff1	mysqld_exporter	🔴 ERROR	0.1 ms	query	SHOW ALL SLAVES STATUS NONBLOCKING
44ebec51	mysqld_exporter	🔴 ERROR	0.1 ms	query	SHOW ALL SLAVES STATUS
b738acf8	mysqld_exporter	🔴 ERROR	0.0 ms	query	SELECT @@query_response_time_stats
4a3cdafc	mysqld_exporter	🔴 ERROR	0.0 ms	query	SHOW ALL SLAVES STATUS
c653dc19	mysqld_exporter	🔴 ERROR	0.0 ms	query	SHOW ALL SLAVES STATUS NOLOCK
9ad7216f	mysqld_exporter	🔴 ERROR	0.1 ms	query	SHOW ALL SLAVES STATUS NONBLOCKING
6662d783	mysqld_exporter	🔴 ERROR	0.0 ms	query	SELECT @@query_response_time_stats
32d588c2	mysqld_exporter	🔴 ERROR	0.1 ms	query	SHOW ALL SLAVES STATUS
60c83d66	mysqld_exporter	🔴 ERROR	0.1 ms	query	SHOW ALL SLAVES STATUS
dc7b8743	mysqld_exporter	🔴 ERROR	0.0 ms	query	SHOW ALL SLAVES STATUS NOLOCK
bbef6a85e	mysqld_exporter	🔴 ERROR	0.1 ms	query	SHOW ALL SLAVES STATUS NONBLOCKING
...

MySQL trace detail

The screenshot shows a monitoring interface with a modal window open. The modal window displays a table of Service Level Objectives (SLOs) and a detailed view of a specific trace span.

Service Level Objective (SLO)

	Objective	Compliance
Availability	99% of requests should not fail ✓	58.3%
Latency	99% of requests should be served faster than 500ms ✓	99.06%

Span: 652e5ddff9e7be16

name	query
service	mysqld
duration	0.03ms
status	ERROR

Attributes

Attribute	Value
container.id	/system.slice/mysqld_exporter.service
db.statement	SHOW ALL SLAVES STATUS
db.system	mysql
host.id	8e0f69d414e64aa69434fab9bc8fe5a9
host.name	db02.esodev.inuits.eu
net.peer.name	127.0.0.1
net.peer.port	3306
otel.scope.name	coroot-node-agent
otel.scope.version	1.28.3
service.name	/system.slice/mysqld_exporter.service

Log Errors

Log Warning

coroot ># ⚡ last hour

search for apps and nodes

instances: 33

expanded (+7 apps)

SLO INSTANCES CPU MEMORY STORAGE NET DNS LOGS PROFILING TRACING

Errors: 1,053 errors occurred
Condition: the number of messages with the ERROR and CRITICAL severity levels > 0

Using container logs (configure)

Container logs

Filter: Debug Info Warning Error Filter messages Query

View: Messages Patterns Newest first Oldest first Limit: 100

warning

Date Message

Nov 15 14:45:07 ts=2024-11-15T13:45:07.155Z caller=dedupe.go:112 component=remote level=warn remote_name=46c9fb url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"
Nov 15 14:43:57 ts=2024-11-15T13:43:57.156Z caller=dedupe.go:112 component=remote level=warn remote_name=708428 url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"
Nov 15 14:43:00 ts=2024-11-15T13:43:00.286Z caller=dedupe.go:112 component=remote level=warn remote_name=46c9fb url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"
Nov 15 14:42:56 ts=2024-11-15T13:42:56.088Z caller=dedupe.go:112 component=remote level=warn remote_name=708428 url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"
Nov 15 14:40:46 ts=2024-11-15T13:40:46.035Z caller=dedupe.go:112 component=remote level=warn remote_name=46c9fb url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"
Nov 15 14:39:29 ts=2024-11-15T13:39:29.523Z caller=dedupe.go:112 component=remote level=warn remote_name=46c9fb url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"
Nov 15 14:39:18 ts=2024-11-15T13:39:18.751Z caller=dedupe.go:112 component=remote level=warn remote_name=708428 url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"
Nov 15 14:37:87 ts=2024-11-15T13:37:07.143Z caller=dedupe.go:112 component=remote level=warn remote_name=46c9fb url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"
Nov 15 14:34:34 ts=2024-11-15T13:34:34.483Z caller=dedupe.go:112 component=remote level=warn remote_name=708428 url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"
Nov 15 14:28:56 ts=2024-11-15T13:28:56.602Z caller=dedupe.go:112 component=remote level=warn remote_name=708428 url=http://olillylab-cortex.olillydev.olilly.eu/api/v1/push msg="Failed to send batch, retrying" err="err"

Alerting

- Per project configuration
- SLO Driven
- Configureable
- Alert integrations

Slack, Teams, PagerDuty, OpsGenie, WebHooks (e.g RocketChat)

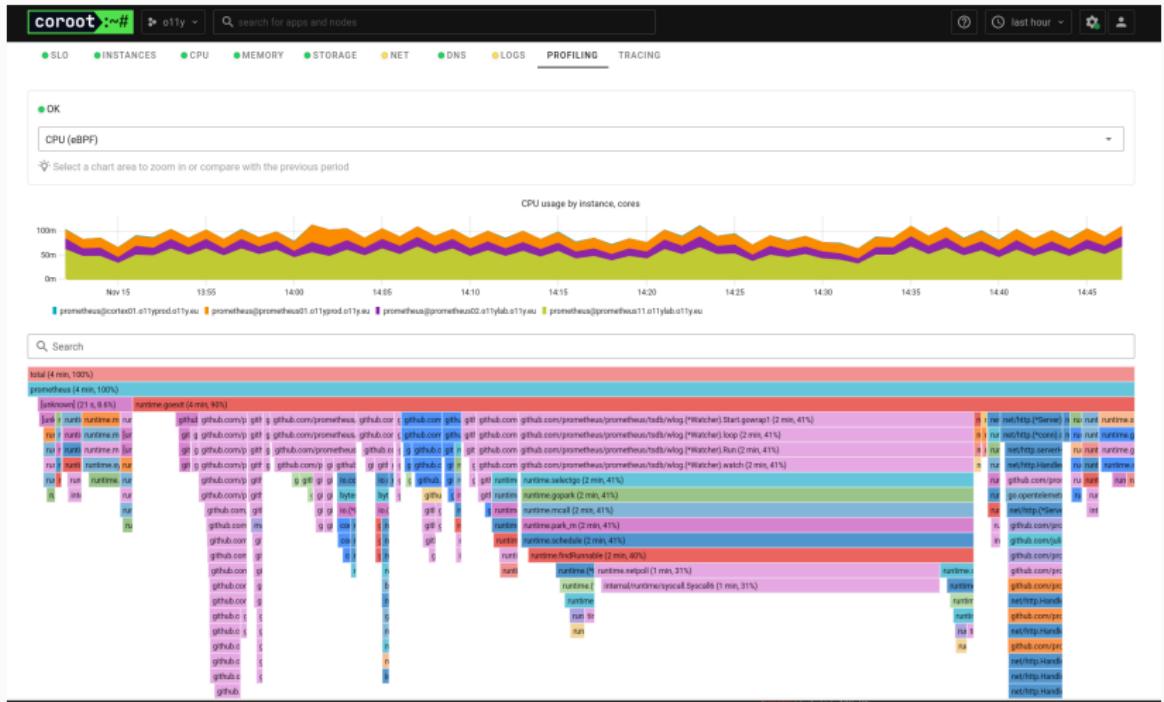
The screenshot shows a Slack channel named '# coroot' with a star icon. The channel has 1,000 members. A message from 'coroot alerts' at 4:47 PM on November 6, 2024, states: '[crowdsec.mgmtprod.inuits.eu:80@external](http://coroot01.o11ylab.o11y.eu:8080/p/d1qrwlw0/incidents?incident=iflmuueb) incident resolved'. Below it, two bullet points indicate SLO status: 'SLO / Availability: error budget burn rate is 0.0x within 1 hour' and 'SLO / Latency: error budget burn rate is 18.3x within 1 hour'. Another message at 7:36 AM on the same day states: 'CRITICAL [crowdsec.mgmtprod.inuits.eu:80@external](http://coroot01.o11ylab.o11y.eu:8080/p/d1qrwlw0/incidents?incident=vzja3m5h) is not meeting its SLOs'. A bullet point below it says 'SLO / Availability: error budget burn rate is 100.0x within 1 hour'. A third message at 8:09 AM states: '[crowdsec.mgmtprod.inuits.eu:80@external](http://coroot01.o11ylab.o11y.eu:8080/p/d1qrwlw0/incidents?incident=vzja3m5h) incident resolved'. Two bullet points below it say 'SLO / Availability: error budget burn rate is 100.0x within 1 hour' and 'SLO / Latency: error budget burn rate is 0.0x within 1 hour'. A fourth message at 8:40 AM states: 'CRITICAL [crowdsec.mgmtprod.inuits.eu:80@external](http://coroot01.o11ylab.o11y.eu:8080/p/d1qrwlw0/incidents?incident=ri4hqesa) is not meeting its SLOs'. A bullet point below it says 'SLO / Availability: error budget burn rate is 100.0x within 1 hour'. A fifth message at 9:13 AM states: 'WARNING [crowdsec.mgmtprod.inuits.eu:80@external](http://coroot01.o11ylab.o11y.eu:8080/p/d1qrwlw0/incidents?incident=ri4hqesa) is not meeting its SLOs'. A bullet point below it says 'SLO / Availability: error budget burn rate is 63.4x within 24 hours'. The bottom of the screenshot shows a footer with the OTTly logo and the number 46.

A word on SLO ..

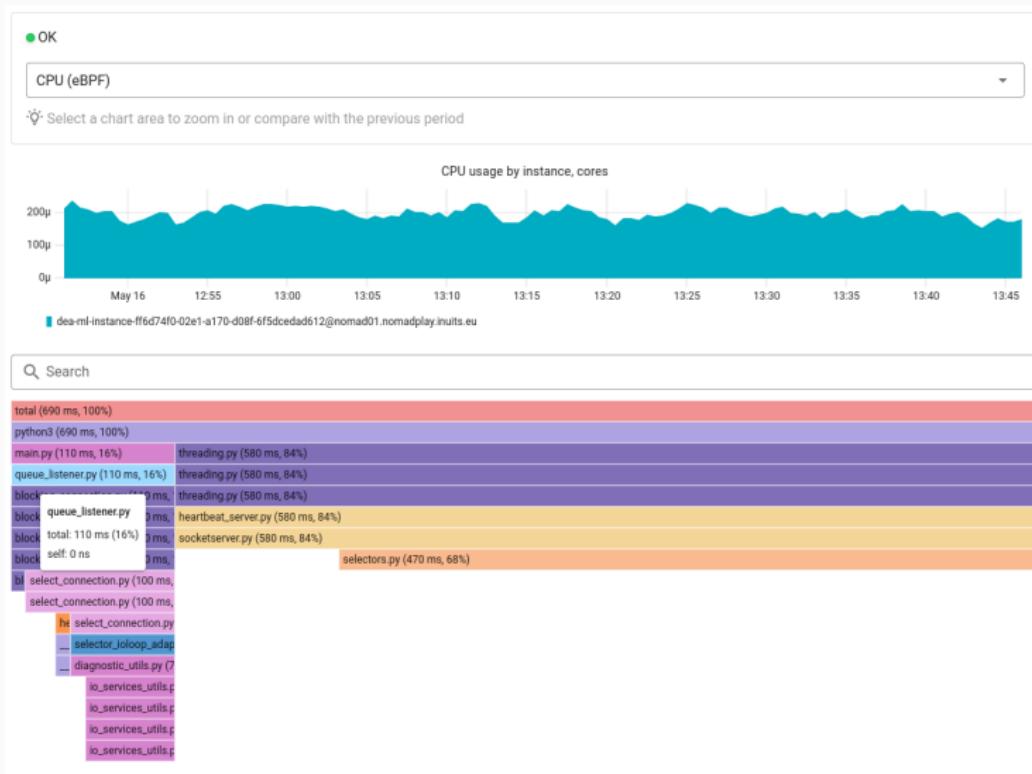
- 100% Fail vs vs 85% available
- Configurable
- Non intrusive alerts are helpful for debugging
- Don't wake people up unless your business is critical
- What if you don't meet them ?
- Are they actionable ?
- Most orgs don't have the right culture
- They don't have their teams empowered

Profiling

- Pyroscope based



Profiling



There is more ..

- Cost calculations
- Deployments
- Otel gateway
- Root Cause Analysis

My personal Use cases today

- Debug Tool
- Informing me where to look
- Fully enabled in dev environments and some prod
- Feature flagged and enabled on demand in others
- Future : enabled everywhere

#o11ylove

I always loved monitoring, done right,

But coroot gave me #o11ylove



Kris Buytaert

@krisbuytaert

kris@o11y.eu

Essensteenweg 31

2930 Brasschaat

Belgium

Contact: info@o11y.eu