Confidence with Chaos for your Kubernetes Observability

Michael Friedrich
Senior Developer Evangelist at GitLab

@dnsmichi

EveryoneCanContribute.com

OSAD
OPEN SOURCE AUTOMATION DAYS

What to expect

Stories from an ex-OSS monitoring maintainer (2009-2020), turned into the Ops in Dev

Metrics, Prometheus, Grafana, kube-prometheus, alerts, SLOs

Chaos with Kubernetes

It's always DNS

Chaos Tracing (or vice versa)

Beyond Observability: OpenTelemetry and eBPF ideas

LEGO - can you spot them all?

OSAD

OPEN SOURCE AUTOMATION DAYS

@dnsmichi

Photo by Michael Friedrich

Kubernetes up and running

Kubernetes

Architecture, Components

Nodes, Pods, Containers

Deployments, Services, …

APIs, ports, data sources

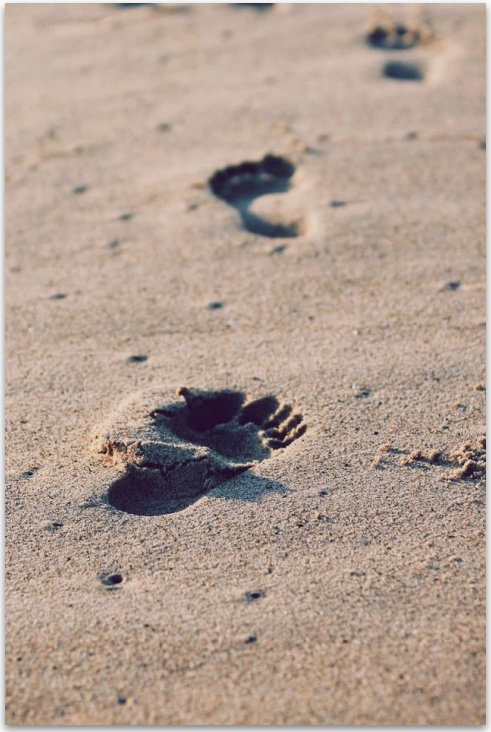


Photo by Christopher Sardegna on Unsplash

We need monitoring

Availability monitoring

Performance and resources

Identify slow or blocking deployments

~~Host-Service State based with microservices~~

Metrics? Logs? More?

Understand all components?

Best practices?

💡 Focus needed

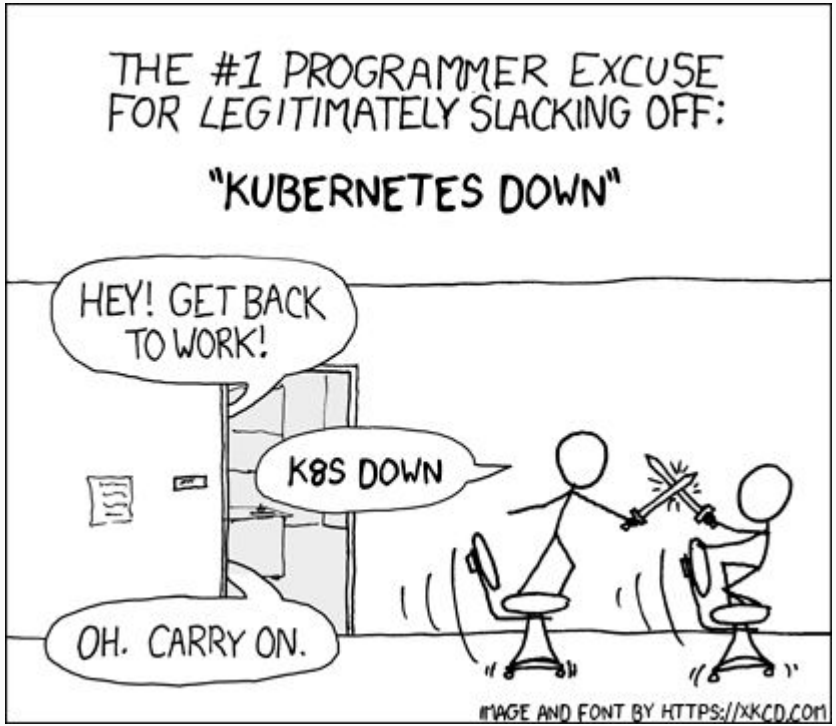


Image generated with https://gitlab.com/gitlab-de/go-excusegen which itself is based on the XKCD comic at https://xkcd.com/303/

Monitoring, first iteration: Metrics

Kubernetes

- Different data sources
  - Service discovery?

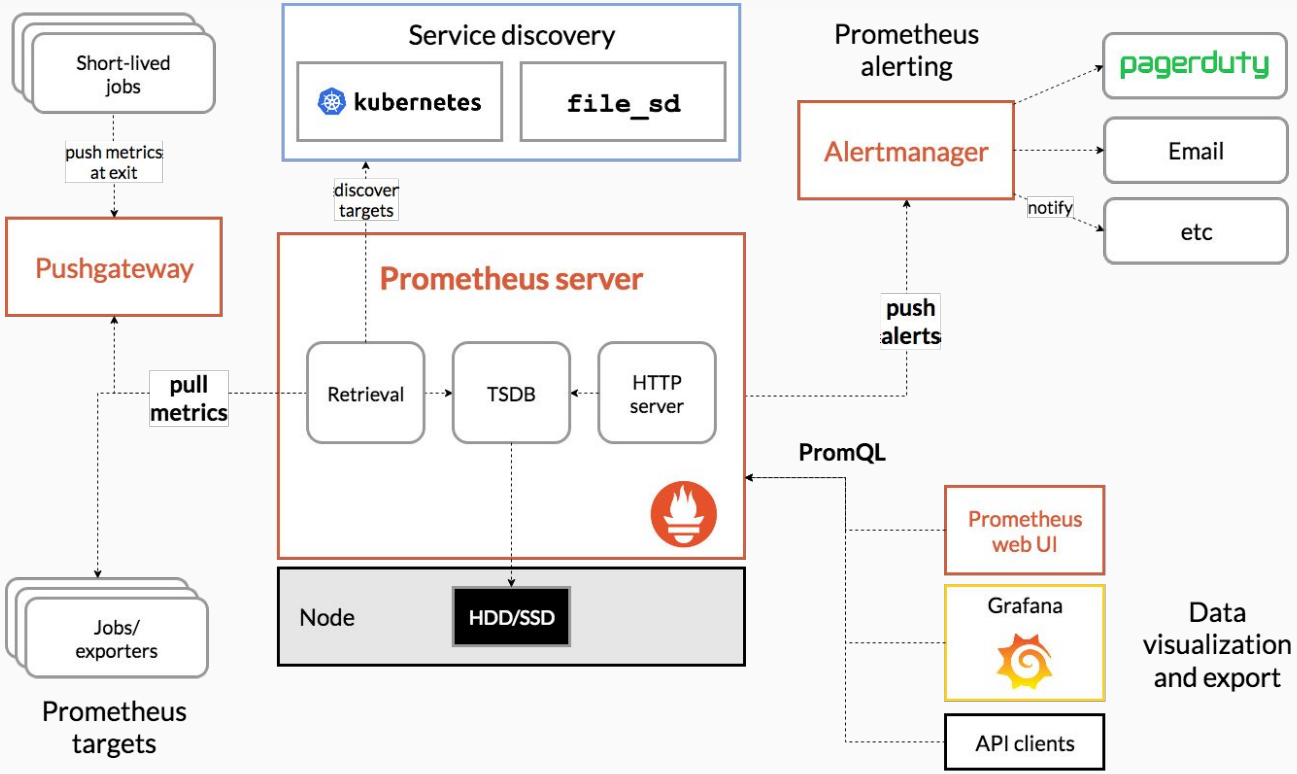
CNCF ecosystem

- Prometheus
  - /metrics
    - TSDB, trends, dashboards
      - SLOs
        - Alerts & incidents

Photo by Valentin G on Unsplash

Prometheus: Architecture

@dnsmichi



Picture from https://prometheus.io/docs/introduction/overview/#architecture

Prometheus: PromQL

"Prometheus provides a functional query language called PromQL (**Prometheus Query Language**) that lets the user select and aggregate time series data in real time.

The result of an **expression** can either be shown as a **graph**, viewed as **tabular data** in Prometheus's expression browser, or **consumed** by external systems via the **HTTP API**."

```
# Latest sample
metric_name

# Range
metric_name[5m]

# Labels
metric_name{label1="a",label
2="b"}

# Functions
rate(metric_name[5m])
sum(metric_name)
delta(metric_name[5m])

# Comparisons
metric_name > 10*1024
```

https://prometheus.io/docs/prometheus/latest/querying/examples/
https://promlabs.com/promql-cheat-sheet/
https://demo.promlens.com/

Prometheus: UI

@dnsmichi

- UI
  - Test scrape targets, query metrics
  - Graphs
- Dashboards in Grafana
  - Prometheus data source, PromQL query helpers
  - Panels and dashboards
- Perses
  - Dashboards as code, GitOps
  - In development at https://github.com/perses/perses

kube-prometheus

kube-prometheus ***deploys the Prometheus Operator*** *and already schedules a* ***Prometheus called prometheus-k8s with alerts and rules by default.*** 😍

Components in a package, otherwise installed manually

- Prometheus Operator, Prometheus (HA), Alert Manager (HA)
- Node exporter, Prometheus Adapter for Kubernetes Metrics API, kube-state-metrics
- Grafana

Pre-configured cluster monitoring

- Grafana dashboards and Alerting mixins

Dynamically create ServiceDiscovery objects

https://prometheus-operator.dev/docs/prologue/quick-start/
https://github.com/prometheus-operator/prometheus-operator#prometheus-operator-vs-kube-prometheus-vs-community-helm-chart

Photo by Márton Szalai on Unsplash

Kubernetes Metrics

Kube-prometheus OOTB

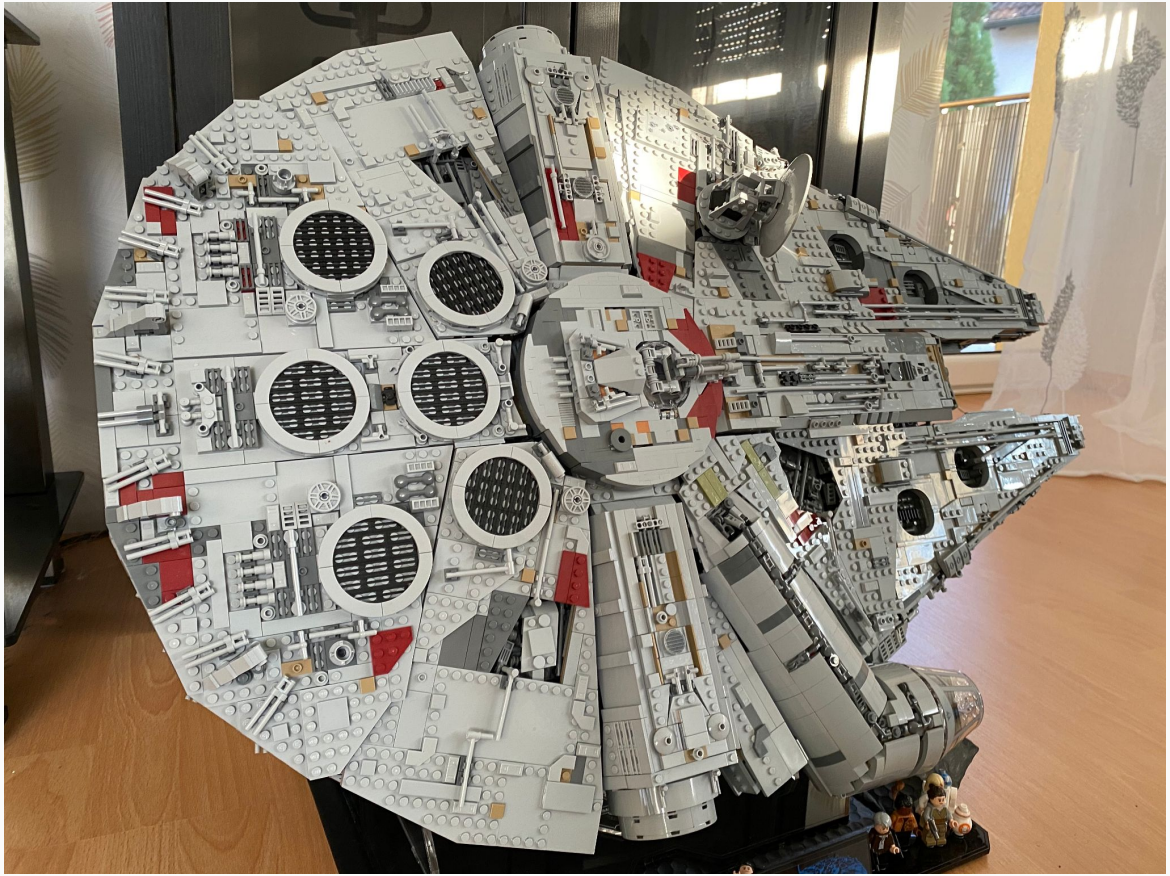


Photo by Michael Friedrich

OSAD
OPEN SOURCE AUTOMATION DAYS

@dnsmichi

Metrics?

@dnsmichi

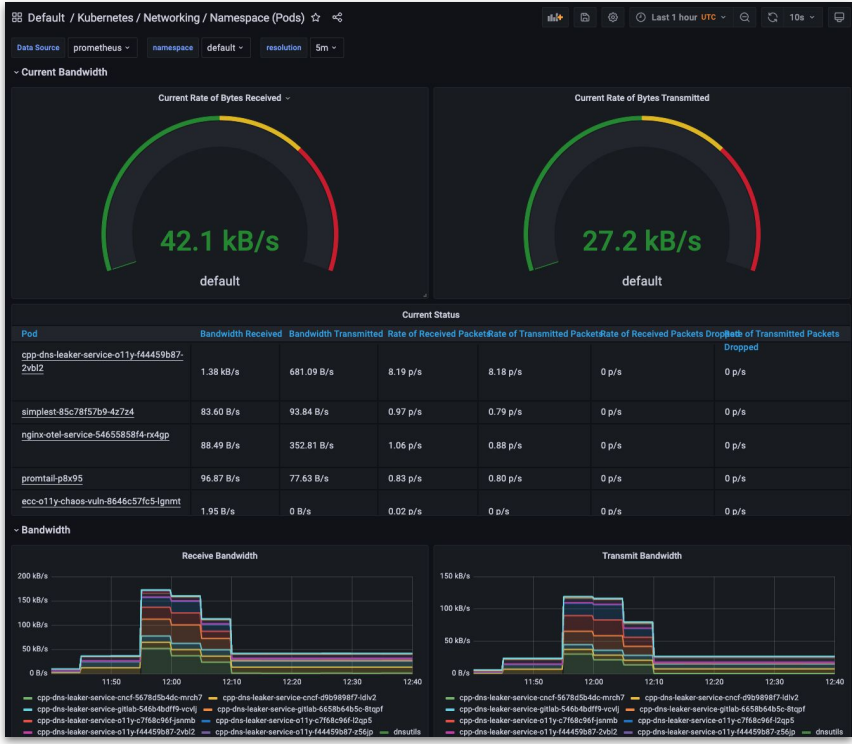
Cluster metrics

- Node status, resource usage
- Deployments, no of pods

Pod metrics

- Kubernetes, containers, applications

So many metrics …

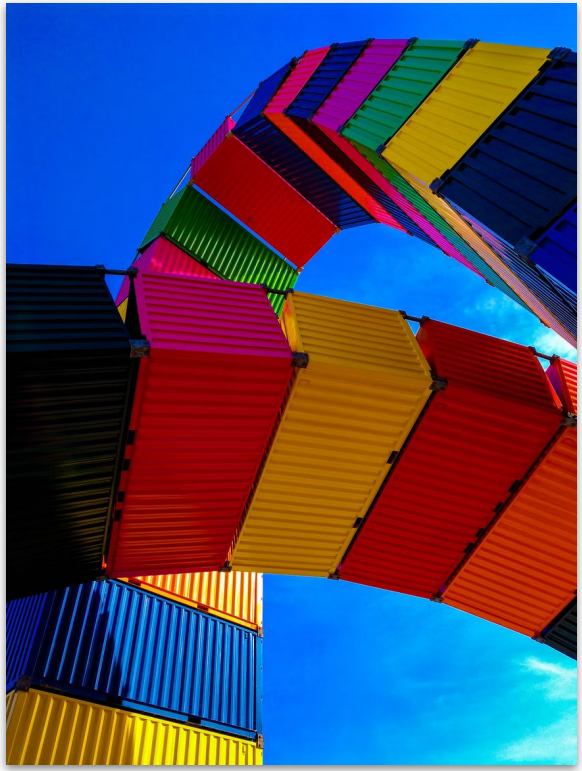


Container Metrics

Kubelets provide embedded cAdvisor exporter at /metrics/cadvisor

Resource usage monitoring, OOM kills, etc.

- `container_cpu_system_seconds_total` - System CPU usage time
- `container_cpu_user_seconds_total` - User CPU usage time
- `container_cpu_usage_seconds_total` - Total CPU usage time (system + user)
- `container_memory_working_set_bytes` - Pod memory usage



https://github.com/google/cadvisor
https://kubernetes.io/docs/concepts/cluster-administration/system-metrics/#metrics-in-kubernetes

Photo by Antoine Petitteville on Unsplash

kube-state-metrics

@dnsmichi

Health of deployments, nodes and pods

Queries the Kubernetes API server

Exposed metrics

- Service
- Deployment
- Pod

Installed with the Prometheus Operator

https://github.com/kubernetes/kube-state-metrics
https://github.com/kubernetes/kube-state-metrics/tree/master/docs#exposed-metrics

Photo by Bekir Dönmez on Unsplash

SLOs and Alerts

Photo by Michael Friedrich

OSAD
OPEN SOURCE AUTOMATION DAYS

@dnsmichi

Metrics are great - what's next?

(what) Definition of failure

(why) Do something because

Reasons (e.g. threshold violated)

Rule matched: Notify and raise alerts

(who) Responsible persona/team

(how) Documentation for incidents, runbooks

(iterate) Corrective actions, Mean time to response (MTTR)

https://about.gitlab.com/handbook/engineering/infrastructure/incident-management/
https://gitlab.com/gitlab-com/runbooks

Photo by Troy Mortier on Unsplash

Alerts with Prometheus

- Prometheus Alert Rules
  - Sends alerts to Alert Manager
- Alert Manager
  - Grouping - similar nature, avoiding duplicated alerts
  - Inhibition - suppress alerts when others are fired already
  - Silences - mute alerts for a given time
- Transports
  - SMTP (Email), Webhook, pushover, SNS
  - Pager (VictorOps, PagerDuty, Opsgenie)
  - Chat (Slack, WeChat, Telegram)

https://prometheus.io/docs/alerting/latest/alertmanager/
https://prometheus.io/docs/alerting/latest/configuration/

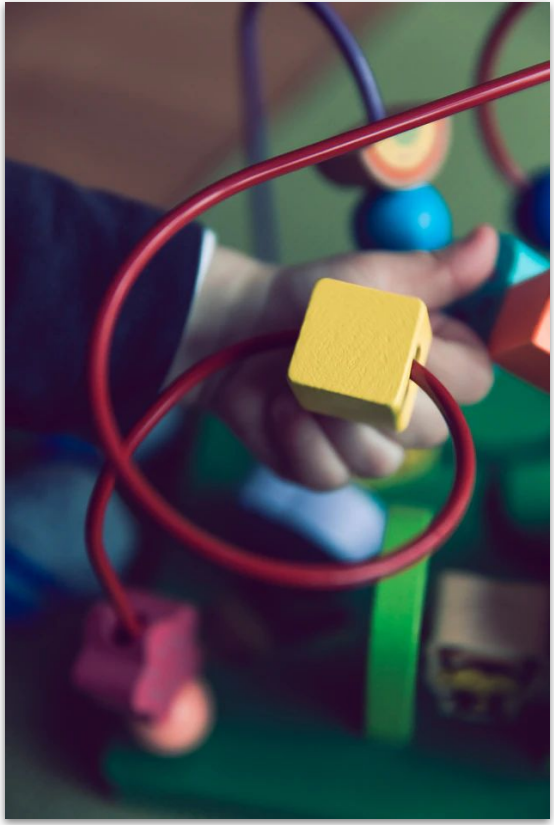


Photo by Markus Spiske on Unsplash

Alerts for Kubernetes

Infrastructure alerts

Memory, CPU, disk pressure

Pods

Error rate / response time

Reachability

```
  - alert: KubernetesPodNotHealthy
    expr: min_over_time(sum by (namespace, pod)
(kube_pod_status_phase{phase=~"Pending|Unknown|Failed"})[15m:1m]
) > 0
    for: 0m
    labels:
      severity: critical
    annotations:
      summary: Kubernetes Pod not healthy (instance {{
$labels.instance }})
      description: "Pod has been in a non-ready state for longer
than 15 minutes.\n  VALUE = {{ $value }}\n  LABELS = {{ $labels
}}"

# Example from
https://awesome-prometheus-alerts.grep.to/rules.html#kubernetes
```

https://awesome-prometheus-alerts.grep.to/rules.html#kubernetes
https://www.zupzup.org/prometheus-alert-examples/

Alert Rules in Prometheus Operator

- PrometheusRule CRD
  - Wraps Prometheus rules format in the operator

```
spec:
  groups:
  - name: kube-state-metrics
    rules:
    - alert: KubeStateMetricsListErrors
      annotations:
        description: kube-state-metrics is experiencing errors at an elevated rate in list operations. This is
likely causing it to not be able to expose metrics about Kubernetes objects correctly or at all.
        runbook_url: https://github.com/prometheus-operator/kube-prometheus/wiki/kubestatemetricslisterrors
        summary: kube-state-metrics is experiencing errors in list operations.
      expr: |
        (sum(rate(kube_state_metrics_list_total{job="kube-state-metrics",result="error"}[5m]))
          /
        sum(rate(kube_state_metrics_list_total{job="kube-state-metrics"}[5m])))
        > 0.01
      for: 15m
      labels:
        severity: critical
```

https://prometheus-operator.dev/docs/operator/api/#prometheusrule
https://prometheus.io/docs/prometheus/latest/configuration/alerting_rules/
https://github.com/prometheus-operator/kube-prometheus/blob/7a3879b/manifests/kube-state-metrics-prometheusRule.yaml

Alert Receivers

Chat: “See something immediately”

Document incidents

- Ticket and issue systems, mailing lists

Refinements and grouping needed

Avoid alert fatigue

- Too many alerts, ignore everything

Dedicated incident and issue management platforms

https://training.promlabs.com/training/alerting-with-prometheus

Photo by Martin Widenka on Unsplash

Break something, get alerts

Trigger alerts manually - kill some pods

Manual work is a bug!

namespace="monitoring" + 17 alerts

2021-06-11T19:06:26.729Z − Info Source Silence

description: Pod monitoring/node-exporter-27p57 has been in a non-ready state for longer than 15 minutes.

runbook_url: https://github.com/prometheus-operator/kube-prometheus/wiki/kubepodnotready

summary: Pod has been in a non-ready state for more than 15 minutes.

alertname="KubePodNotReady" + pod="node-exporter-27p57" + prometheus="monitoring/k8s" + severity="warning" +

Break more in a fun way

@dnsmichi

Kubedoom (might not be applicable for $work)

```
kubectl logs -f kubedoom-6f8c65fd85-52fx2 -n kubedoom
Demon: default/cpp-dns-leaker-service-o11y-55d9498b7f-6jdng, 1604833570
Demon: default/cpp-dns-leaker-service-cncf-75d756dcdf-9txkj, 536154357
Demon: default/cpp-dns-leaker-service-gitlab-5ddf58f457-cgwrs, 1376373469
Demon: default/simplest-85c78f57b9-54nrw, 1630267728
Demon: default/cpp-dns-leaker-service-o11y-55d9498b7f-lmrr8, 1668996446
Demon: default/ecc-o11y-chaos-vuln-6c86794899-jr5zk, 1001572467
   process 994606398 [11y-chaos-vuln-6c86794899-plrrk] monster removed from engine
kill process 1001572467
2022/06/23 19:25:19 Pod to kill: default/ecc-o11y-chaos-vuln-6c86794899-jr5zk
kill process 1376373469
2022/06/23 19:25:20 Pod to kill: default/cpp-dns-leaker-service-gitlab-5ddf58f457-cgwrs
***** game ticker: *****
Demon: default/cpp-dns-leaker-service-o11y-55d9498b7f-6jdng, 1604833570
Demon: default/cpp-dns-leaker-service-cncf-75d756dcdf-9txkj, 536154357
Demon: default/simplest-85c78f57b9-54nrw, 1630267728
Demon: default/cpp-dns-leaker-service-o11y-55d9498b7f-lmrr8, 1668996446
Demon: default/ecc-o11y-chaos-vuln-6c86794899-jr5zk, 1001572467
   process 1001572467 [11y-chaos-vuln-6c86794899-jr5zk] monster resurrect
Demon: default/ecc-o11y-chaos-vuln-6c86794899-dj2q4, 1008979108
   process 1008979108 [default/ecc-o11y-chaos-vuln-6c86794899-dj2q4] monster at 1960 -3280
Demon: default/cpp-dns-leaker-service-gitlab-5ddf58f457-m6z47, 1366274011
   process 1366274011 [default/cpp-dns-leaker-service-gitlab-5ddf58f457-m6z47] monster at 2240 -356
0
```

https://gitlab.com/everyonecancontribute/observability/k8s-o11y-chaos#game-kubedoom

Simulate incidents?

Simulating production incidents is hard 🤔

Add automated chaos 💡

- Staging environments
- Production deployments

Trigger alerts and SLOs 🔥

Photo by Matteo Di Iorio on Unsplash

Observability and Chaos Engineering

Chaos Experiments

Photo by Michael Friedrich

OSAD

OPEN SOURCE AUTOMATION DAYS

@dnsmichi

As a SRE/Dev/Ops/DevOps, where to?

@dnsmichi

Cloud Native

Clusters

Deployments

Chaos Framework

Experiments

Instrumentation SDKs

https://o11y.love/topics/chaos-engineering/

Chaos Mesh

@dnsmichi

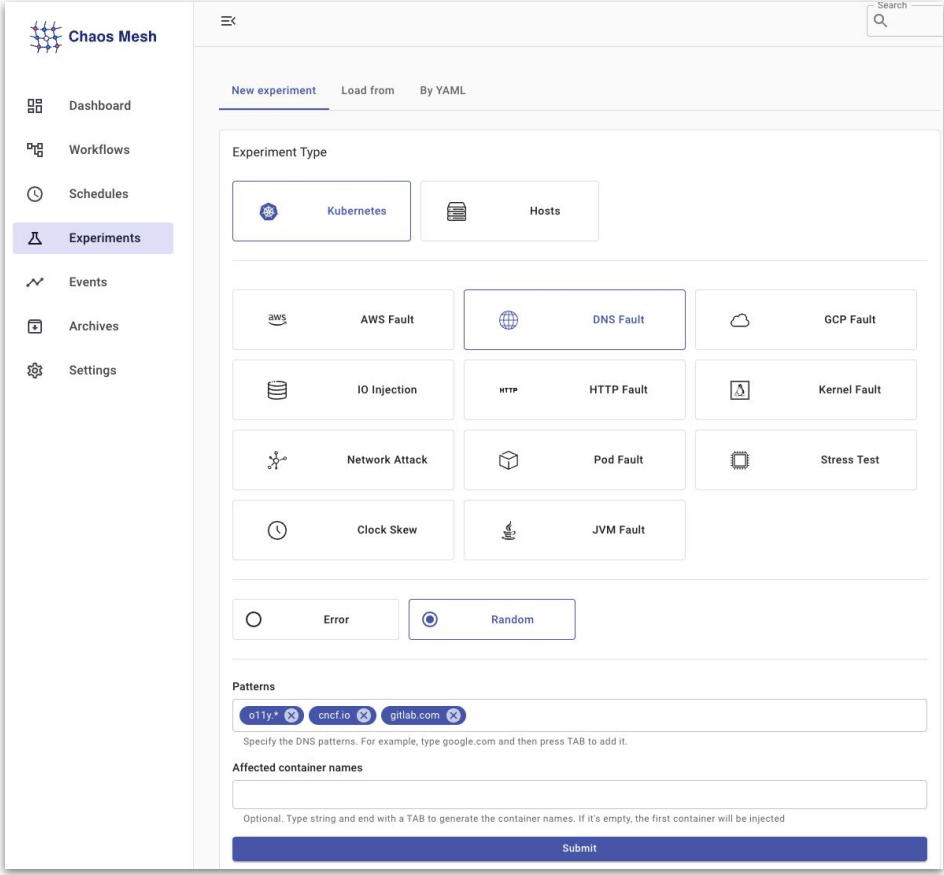
Fail Kubernetes or Hosts

Pods, network, http, time, DNS, ...

Run Chaos Experiments

Once

Continuously as schedule



https://chaos-mesh.org/docs/quick-start/
https://chaos-mesh.org/docs/simulate-dns-chaos-on-kubernetes/

Generate some chaos

@dnsmichi

pod-fail-schedule.yaml 440 bytes

```yaml
kind: Schedule
apiVersion: chaos-mesh.org/v1alpha1
metadata:
  namespace: default
  name: pod-kill-cpp-dns-leaker
spec:
  schedule: '* * * * * '
  startingDeadlineSeconds: null
  concurrencyPolicy: Forbid
  historyLimit: 1
  type: PodChaos
  podChaos:
    selector:
      namespaces:
        - default
      labelSelectors:
        app: cpp-dns-leaker-service-o11y
    mode: all
    action: pod-failure
    duration: 10s
    gracePeriod: 0
```

```
~/d/e/o/k8s-o11y-chaos/k/chaos  main !1 ?5  ✔  system Node  ecc-o11y-chaos ○  09:52:42 PM
kubectl get pods
NAME                                          READY   STATUS             RESTARTS        AGE
simplest-85c78f57b9-4z7z4                     1/1     Running            1 (6d10h ago)   8d
cpp-dns-leaker-service-gitlab-546b4bdff9-lvsn8 1/1    Running            0               41d
ecc-o11y-chaos-vuln-8646c57fc5-brzww          1/1     Running            0               41d
cpp-dns-leaker-service-cncf-d9b9898f7-zq6gp   1/1     Running            0               41d
testcurl                                      1/1     Running            0               25h
nginx-otel-service-c9c56c87c-bz78r            1/1     Running            0               24h
promtail-ncckw                                1/1     Running            1 (31h ago)     49d
dnsutils                                      1/1     Running            25 (11m ago)    25h
promtail-p4f55                                0/1     Running            1 (6d10h ago)   49d
promtail-p8x95                                0/1     Running            0               49d
cpp-dns-leaker-service-o11y-f44459b87-klz8m   0/1     CrashLoopBackOff   7 (2s ago)      39d
cpp-dns-leaker-service-o11y-f44459b87-2vbl2   0/1     CrashLoopBackOff   8 (2s ago)      8d
```

Is this the “correct” chaos?

(we can see that Kubernetes heals itself)

https://gitlab.com/everyonecancontribute/observability/k8s-o11y-chaos/-/blob/main/kubernetes/chaos/pod-fail-schedule.yaml
https://songrgg.github.io/operation/how-to-alert-for-Pod-Restart-OOMKilled-in-Kubernetes/

DevOps Chaos

Real world chaos

OSAD
OPEN SOURCE AUTOMATION DAYS

@dnsmichi

Photo by Michael Friedrich

It’s always DNS

@dnsmichi

Use case: DNS errors cause memory leaks

App creates receive buffer

DNS resolve fails, or empty result

Memory leak, missing buffer delete

```
while (true) {
    // pre-allocated receive buffer
    char *rcvbuf  = new char[1024*1024];

    boost::asio::ip::tcp::resolver::results_type results = resolver.resolve(hostname, "443", error);

    // DNS resolution failed or has empty results, handle the error
    if (error != boost::system::errc::success || results.empty()) {
        handle_dns_error(error);
        continue;
    }
```

https://gitlab.com/everyonecancontribute/observability/cpp-dns-leaker
More details in my Kubecon EU talk: https://www.youtube.com/watch?v=BkREMg8adaI

Image generated with https://gitlab.com/gitlab-de/go-excusegen which itself is based on the XKCD comic at https://xkcd.com/303/

Boring DNS Chaos: 0 replicas for coredns

@dnsmichi

"My own chaos" – Niclas Mietz in the 52. #EveryoneCanContribute cafe

We talked about failing DNS resolving and see how apps behave

```
$ kubectl scale --replicas 0 deploy/coredns -n kube-system

$ kubectl scale --replicas 2 deploy/coredns -n kube-system
```

```
Resolve host 'o11y.love', got 1 results.
35.185.44.232:443
Resolve host 'o11y.love', got 1 results.
35.185.44.232:443
Resolve host 'o11y.love', got 1 results.
35.185.44.232:443
Resolve host 'o11y.love', got 1 results.
35.185.44.232:443
Host not found (non-authoritative), try again later
Host not found (non-authoritative), try again later
Host not found (non-authoritative), try again later
Host not found (non-authoritative), try again later
Host not found (non-authoritative), try again later
Host not found (non-authoritative), try again later
Host not found (non-authoritative), try again later
Host not found (non-authoritative), try again later
```

https://everyonecancontribute.com/post/2022-06-16-cafe-52-learned-at-kubecon-eu-coffee-chat/

Controlled DNS Chaos with Chaos Mesh

@dnsmichi

Definition

```yaml
kind: Schedule
apiVersion: chaos-mesh.org/v1alpha1
metadata:
  namespace: default
  name: schedule-dns-chaos-demo-open-source-automation-days
spec:
  schedule: '* * * * *'
  startingDeadlineSeconds: null
  concurrencyPolicy: Allow
  historyLimit: 2
  type: DNSChaos
  dnsChaos:
    action: error
    selector:
      namespaces:
        - default
    mode: all
    duration: 60s
    patterns:
      - o11y.*
      - cncf.io
      - gitlab.com
      - containerdays.io
      - osad-munich.org
```

Preview of Pods to be injected

Checking or unchecking Pods to further limit the scope.

	Name	Namespace	IP address
☑	cpp-dns-leaker-service-open-source-automation-days-7858cb8mqjhg	default	10.42.0.46

https://gitlab.com/everyonecancontribute/observability/k8s-o11y-chaos/-/tree/main/kubernetes/chaos
https://opensource.com/article/21/6/chaos-mesh-kubernetes

Demo 🙃

Demo hosted by Civo

DNS Memory Leaks: SLO?

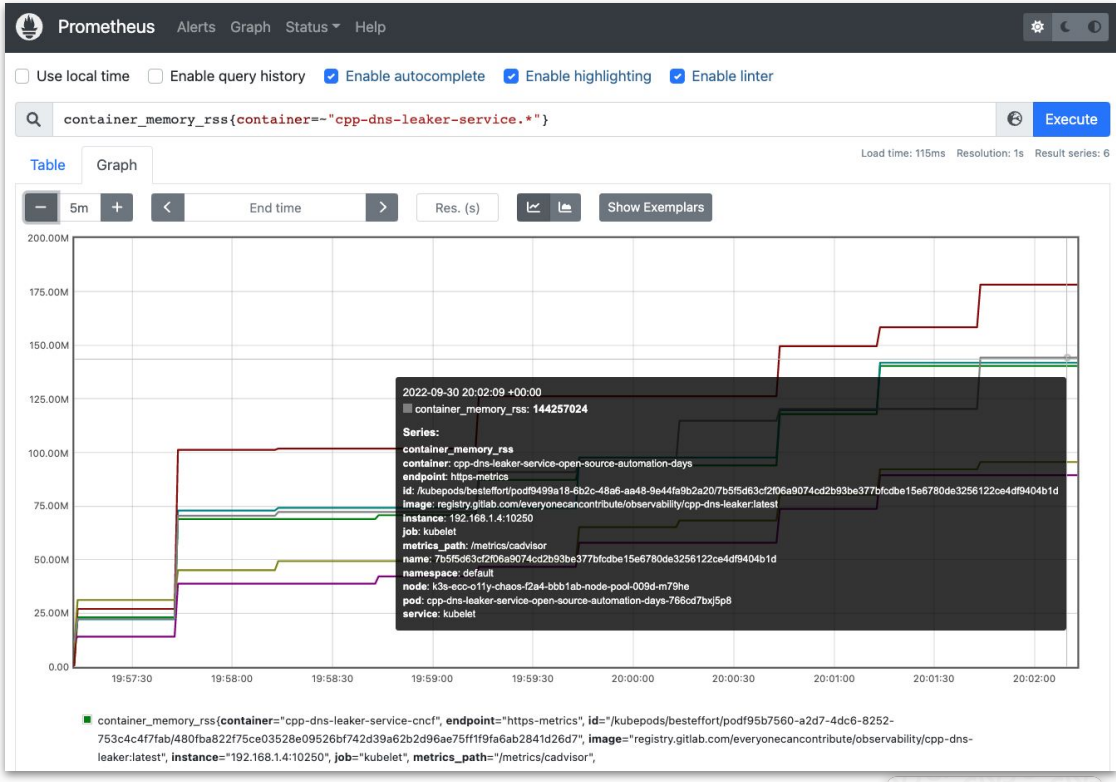
@dnsmichi

SLI: Container Memory Usage

SLO: 10MB

Chaos: DNS

Alert: everyone?



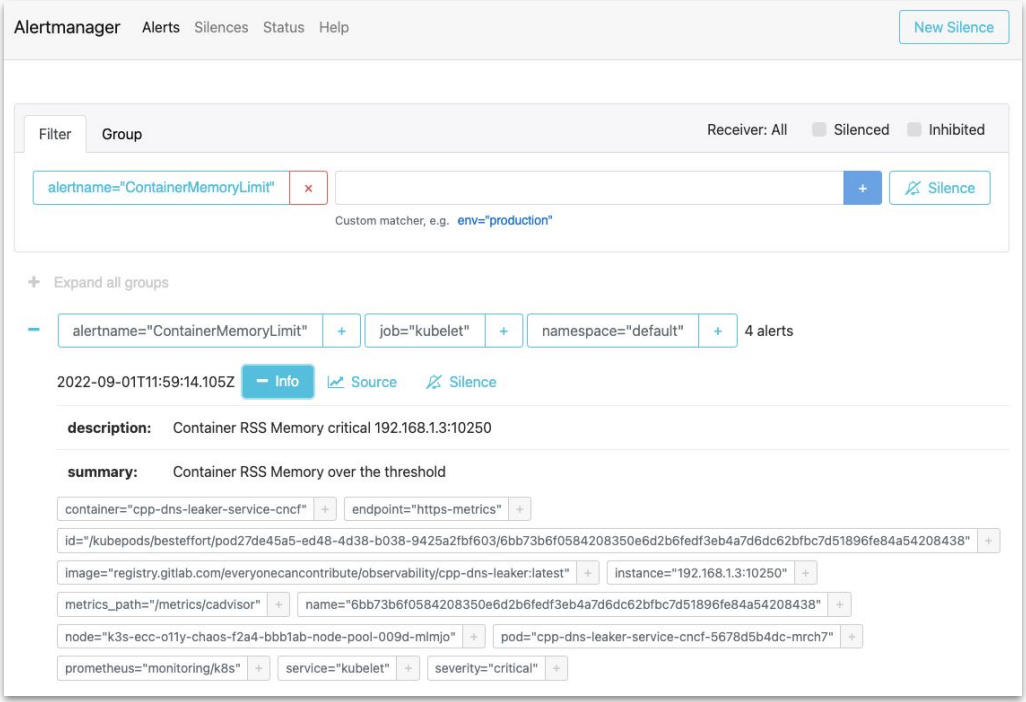
https://gitlab.com/everyonecancontribute/observability/cpp-dns-leaker
More details in my Kubecon EU talk: https://www.youtube.com/watch?v=BkREMg8adaI

Demo hosted by Civo

Alerts

@dnsmichi

```yaml
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  labels:
    app: kube-prometheus-stack
  name: container-rss-memory-alerts
  namespace: monitoring
spec:
  groups:
  - name: critical-rules
    rules:
    - alert: ContainerMemoryLimit
      expr:
container_memory_rss{container=~"cpp-dns-leaker-servi
ce.*"} > 10*1024*1024
      # Use short trigger time for the demo
      for: 1m
      labels:
        severity: critical
      annotations:
        summary: Container RSS Memory over the
threshold
        description: "Container RSS Memory critical\n
{{ $labels.instance }}"
```



https://gitlab.com/everyonecancontribute/observability/k8s-o11y-chaos/-/blob/main/kubernetes/o11y/my-kube-prometheus/alerts/cpp-dns-leaker-memory.yml

Demo hosted by Civo

Chaos generated alerts and red dashboards

- Whats next?
- Alert counts
  - Grouping?
  - Additional context in alerts
- More focus in dashboards
  - Context with URLs & docs
  - Correlate data
  - Reduce the amount of visible data

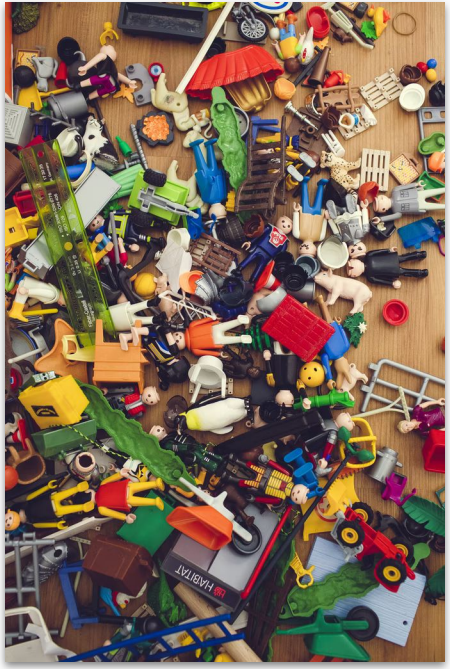


Photo by Markus Spiske on Unsplash

OSAD
OPEN SOURCE AUTOMATION DAYS

@dnsmichi

From metrics to SLOs

Monitoring

- Metrics
  - key/tag
    - Values

Alerts on PromQL queries

- SLOs

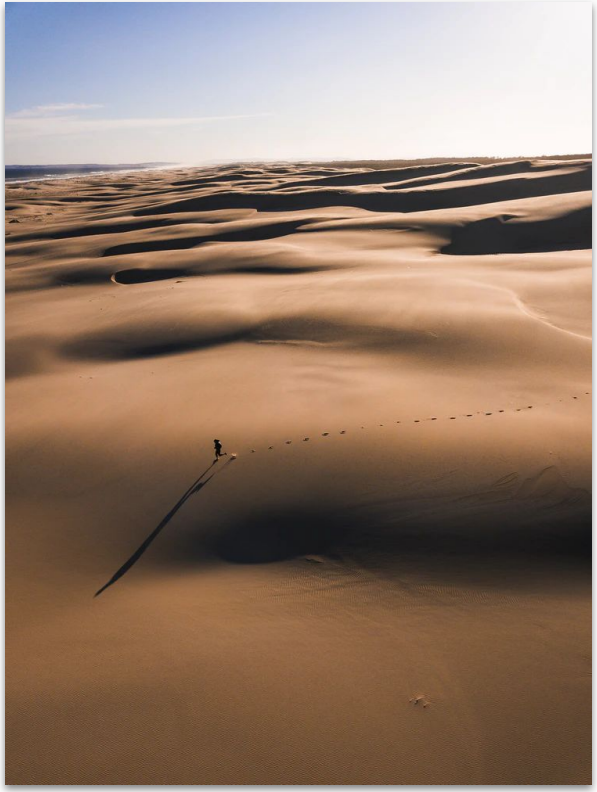


Photo by Silas Baisch on Unsplash

Ops SLOs: Start with Golden Signals

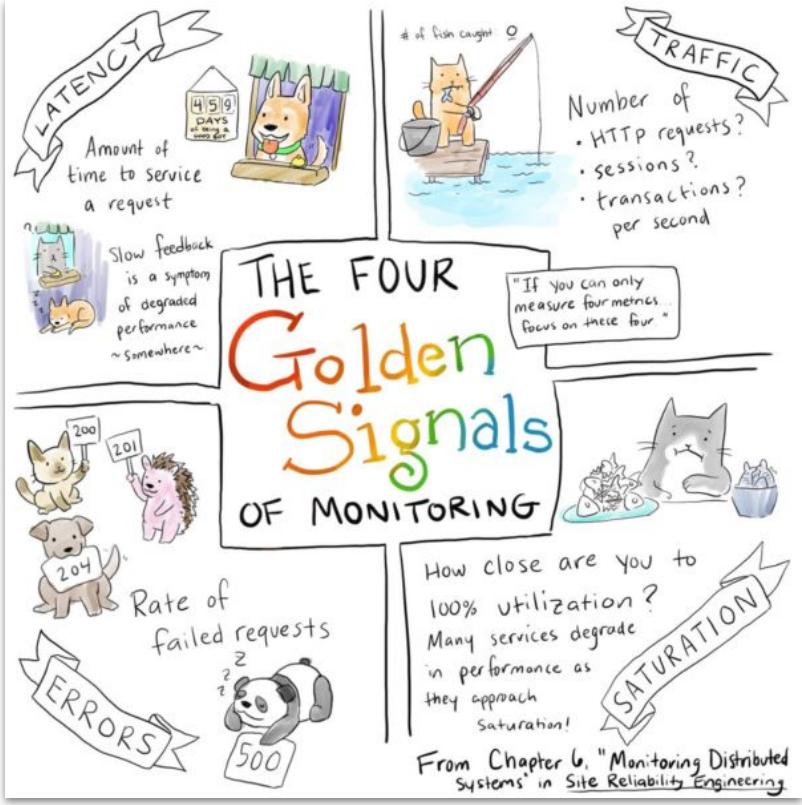
@dnsmichi

Latency

Traffic

Errors

Saturation



https://sysdig.com/blog/golden-signals-kubernetes/
https://sre.google/books/

Image credit: Denise Yu

Ops Iteration

- Many dashboards & alerts
  - Learn & Document
    - Customizations
    - Onboarding
- Goals
  - *Immediately see what's important during an incident*
  - *Reduce MTTR (mean-time-to-response)*

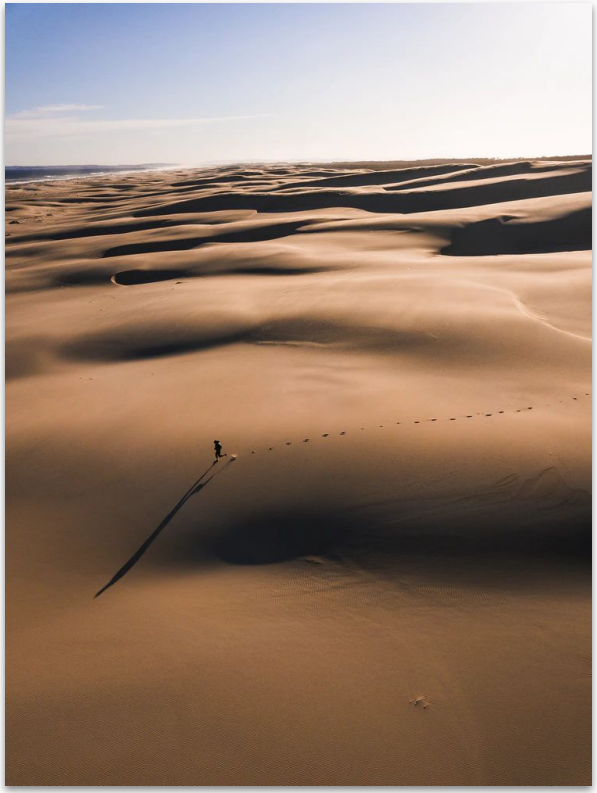


Photo by Silas Baisch on Unsplash

Customize kube-prometheus

@dnsmichi

Use jsonnet

Develop rules and dashboards

Monitor other namespaces

Add applications

customize.jsonnet — my-kube-prometheus

customize.jsonnet

customize.jsonnet

```
local kp = (import 'kube-prometheus/main.libsonnet') + {
  _config+:: {
    namespace: 'monitoring',

    prometheus+:: {
      namespaces: ["default", "kube-system", "ecc"],
    },
  },
};

{ ['00namespace-' + name]: kp.kubePrometheus[name] for name in std.objectFields(kp.kubePrometheus) } +
{ ['0prometheus-operator-' + name]: kp.prometheusOperator[name] for name in std.objectFields(kp.prometheusOperator) } +
{ ['node-exporter-' + name]: kp.nodeExporter[name] for name in std.objectFields(kp.nodeExporter) } +
{ ['kube-state-metrics-' + name]: kp.kubeStateMetrics[name] for name in std.objectFields(kp.kubeStateMetrics) } +
{ ['alertmanager-' + name]: kp.alertmanager[name] for name in std.objectFields(kp.alertmanager) } +
{ ['prometheus-' + name]: kp.prometheus[name] for name in std.objectFields(kp.prometheus) } +
{ ['grafana-' + name]: kp.grafana[name] for name in std.objectFields(kp.grafana) }
```

https://prometheus-operator.dev/docs/developing-prometheus-rules-and-grafana-dashboards/
https://prometheus-operator.dev/docs/kube/monitoring-other-namespaces/
https://github.com/prometheus-operator/kube-prometheus#customizing-kube-prometheus
https://docs.bitnami.com/tutorials/implementing-slos-using-prometheus

Custom dashboards

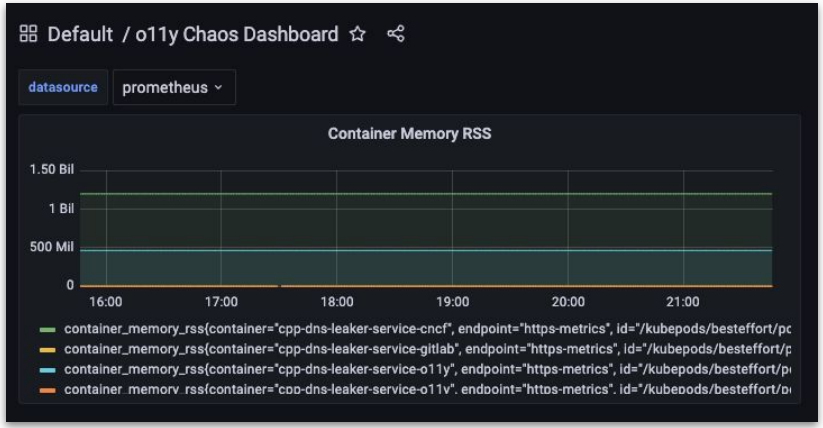
@dnsmichi

o11y-chaos-dashboard.jsonnet 857 bytes

```
local grafana = import 'grafonnet/grafana.libsonnet';
local dashboard = grafana.dashboard;
local row = grafana.row;
local prometheus = grafana.prometheus;
local template = grafana.template;
local graphPanel = grafana.graphPanel;

{
  'o11y-chaos-dashboard.json':
    dashboard.new('o11y Chaos Dashboard')
    .addTemplate(
      {
        current: {
          text: 'Prometheus',
          value: 'Prometheus',
        },
        hide: 0,
        label: null,
        name: 'datasource',
        options: [],
        query: 'prometheus',
        refresh: 1,
        regex: '',
        type: 'datasource',
      },
    )
    .addRow(
      row.new()
      .addPanel(graphPanel.new('Container Memory RSS', span=6, datasource='$datasource')
                .addTarget(prometheus.target('container_memory_rss{container=~"cpp-dns-leaker-service.*"}')))
    ),
}
```

```
31      grafana+: {
32        dashboards+:: (import 'o11y-chaos-dashboard.jsonnet'),
33      },
```



https://prometheus-operator.dev/docs/developing-prometheus-rules-and-grafana-dashboards/
https://github.com/prometheus-operator/kube-prometheus#customizing-kube-prometheus
https://gitlab.com/everyonecancontribute/observability/k8s-o11y-chaos/-/blob/main/kubernetes/o11y/my-kube-prometheus/o11y-chaos-dashboard.jsonnet

Monitor deployed applications with /metrics endpoint

@dnsmichi

- Prometheus server scrapes HTTP endpoints with /metrics
  - Prometheus Exporters (Node, Blackbox, etc.)
  - Instrumented apps
- Prometheus Operator provides ServiceMonitor CRD
  - Define target Endpoints
  - Auto-discover targets from different namespaces

TL;DR: Provide ServiceMonitor CRD in your app deployments.

https://prometheus-operator.dev/docs/operator/design/#servicemonitor

Photo by Bernard Hermant on Unsplash

Beyond Metrics

Photo by Michael Friedrich

OSAD
OPEN SOURCE AUTOMATION DAYS

@dnsmichi

Wait, more than metrics?

@dnsmichi

Metrics

Logs/Events

Traces

Profiling

Error tracking

<more event types>

… is it Observability?

https://about.gitlab.com/blog/2022/06/14/observability-vs-monitoring-in-devops/

Photo by Eric Masur on Unsplash

Logs - too many decisions

Central log management evaluation

Self-managed

- Elastic/OpenSearch, Beats agents
- Grafana Loki/promtail
- Fluentbit/vector as sidecars

… or SaaS

- Datadog, Splunk, Dynatrace, etc.

What logs do we really need?

https://blog.flant.com/logs-in-kubernetes-expectations-vs-reality/

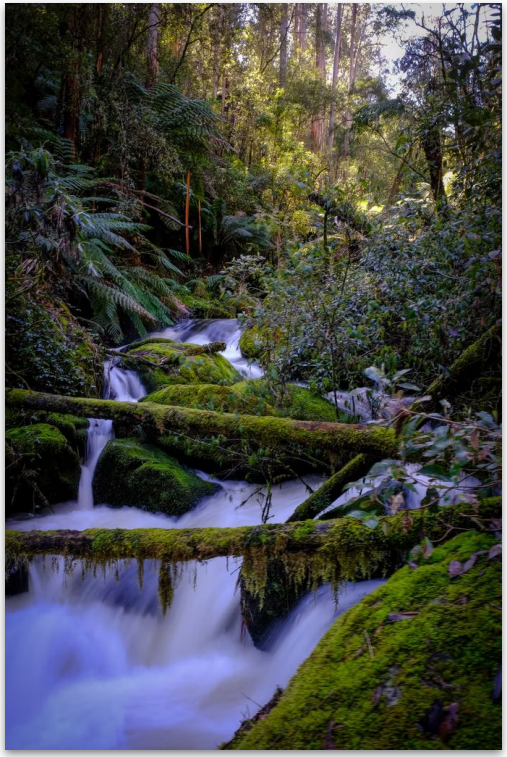
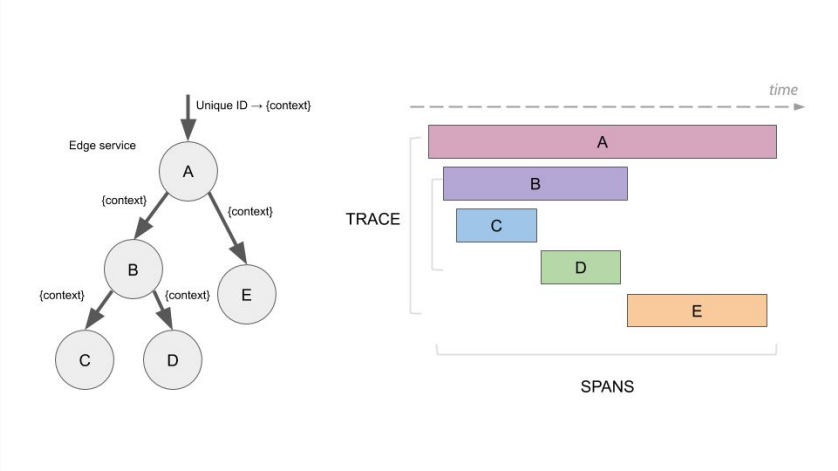


Photo by Zac Porter on Unsplash

Tracing

- Traces, like logs
  - Spans with start/end time
  - Context
  - Metadata enrichment
- Instrumentation
  - Code changes by developers
  - Automated
- Distributed environments



https://www.jaegertracing.io/docs/1.25/architecture/
https://github.com/grafana/tempo/tree/main/example/helm
https://github.com/open-telemetry/opentelemetry-operator
https://docs.google.com/presentation/d/1MAVFeSsTNVWC9wPGOlg83wh8GFtR9hPbdVHuumtgOWA/edit

Image credit: Jaeger Tracing Docs

OpenTelemetry

@dnsmichi

Specification and framework beyond vendors

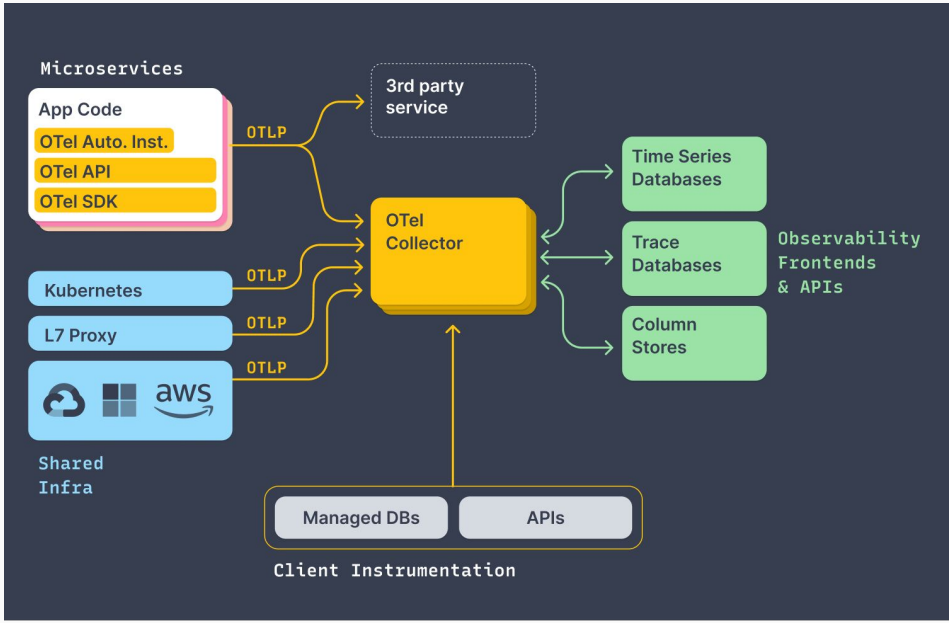
Collector + Sidecar

Bring your own backend

- Jaeger for Traces
- Prometheus for Metrics

Client libraries & SDKs

- Manual instrumentation (C++, Go, etc.)
- Auto-instrumentation (Java)



https://opentelemetry.io/docs/instrumentation/go/getting-started/

Image credit: OpenTelemtry Docs

Focus: Traces

- Kubernetes
  - Components send traces
  - App instrumentation with traces
    - Send to OpenTelemetry collector
      - Store in Jaeger, etc.
        - Visualize, correlate, alert

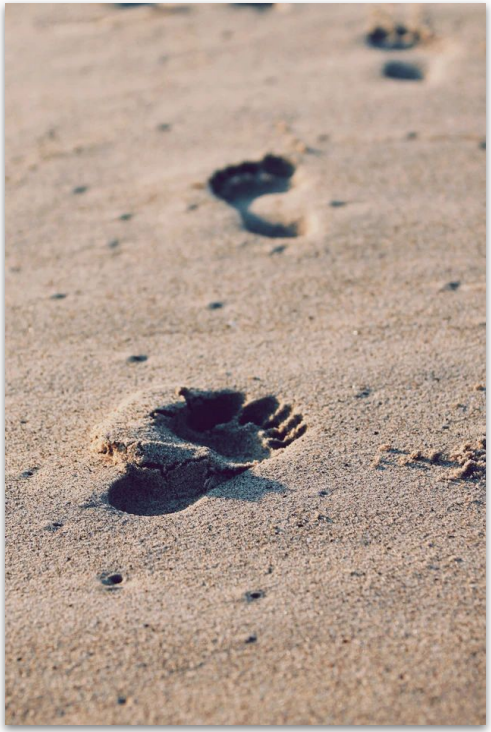


Photo by Christopher Sardegna on Unsplash

Traces help analyse slow requests

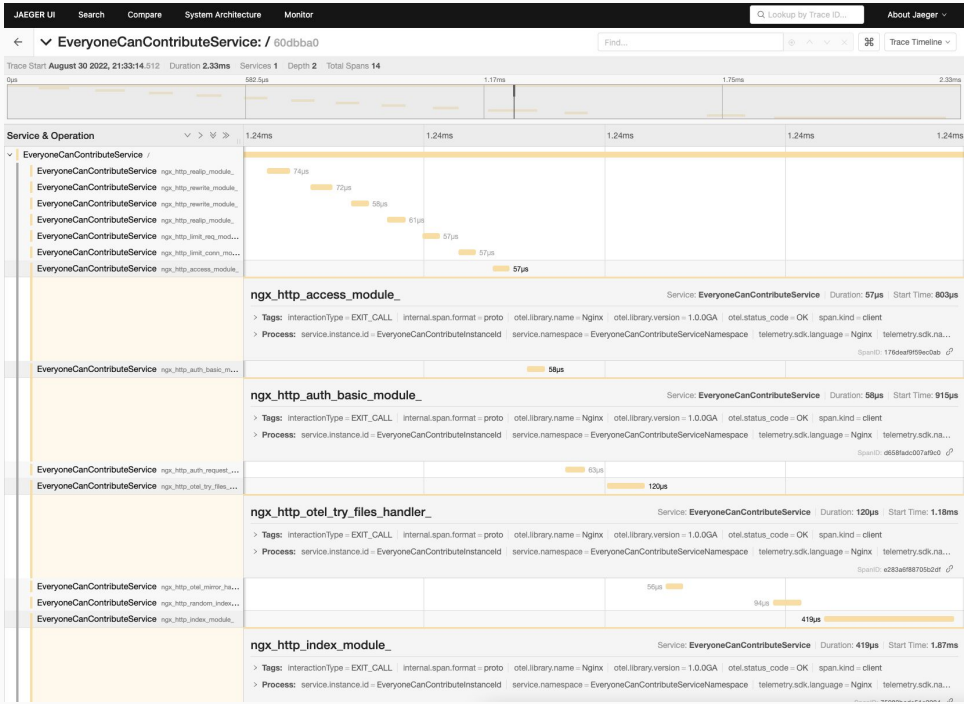
@dnsmichi

Client requests data from HTTP server

Server asks backend, collects data, creates client response

Instrument Nginx/Apache to see traces

OpenTelemetry webserver SDK



https://gitlab.com/everyonecancontribute/observability/nginx-opentelemetry

Traces show slow requests: Chaos

@dnsmichi

How to get slow HTTP requests?

Add `sleep(rand(10))` to your code and deploy … to prod with a feature flag?

Maybe. If you are crazy like me (code reviews will reject the idea ;-))

Better: Chaos experiments for HTTP, network, CPU/Memory stress tests

https://chaos-mesh.org/docs/simulate-http-chaos-on-kubernetes/
https://chaos-mesh.org/docs/simulate-heavy-stress-on-kubernetes/

stress-cpu-mem-nginx-otel.yml 514 bytes

```yaml
kind: Schedule
apiVersion: chaos-mesh.org/v1alpha1
metadata:
  namespace: default
  name: stress-cpu-mem-nginx-otel
spec:
  schedule: '* * * * * '
  startingDeadlineSeconds: null
  concurrencyPolicy: Forbid
  historyLimit: 1
  type: StressChaos
  stressChaos:
    selector:
      namespaces:
        - default
      labelSelectors:
        app: nginx-otel-service
    mode: all
    stressors:
      memory:
        workers: 10
        size: '100'
      cpu:
        workers: 10
        load: 100
    duration: 60s
```

Traces show slow requests: Chaos

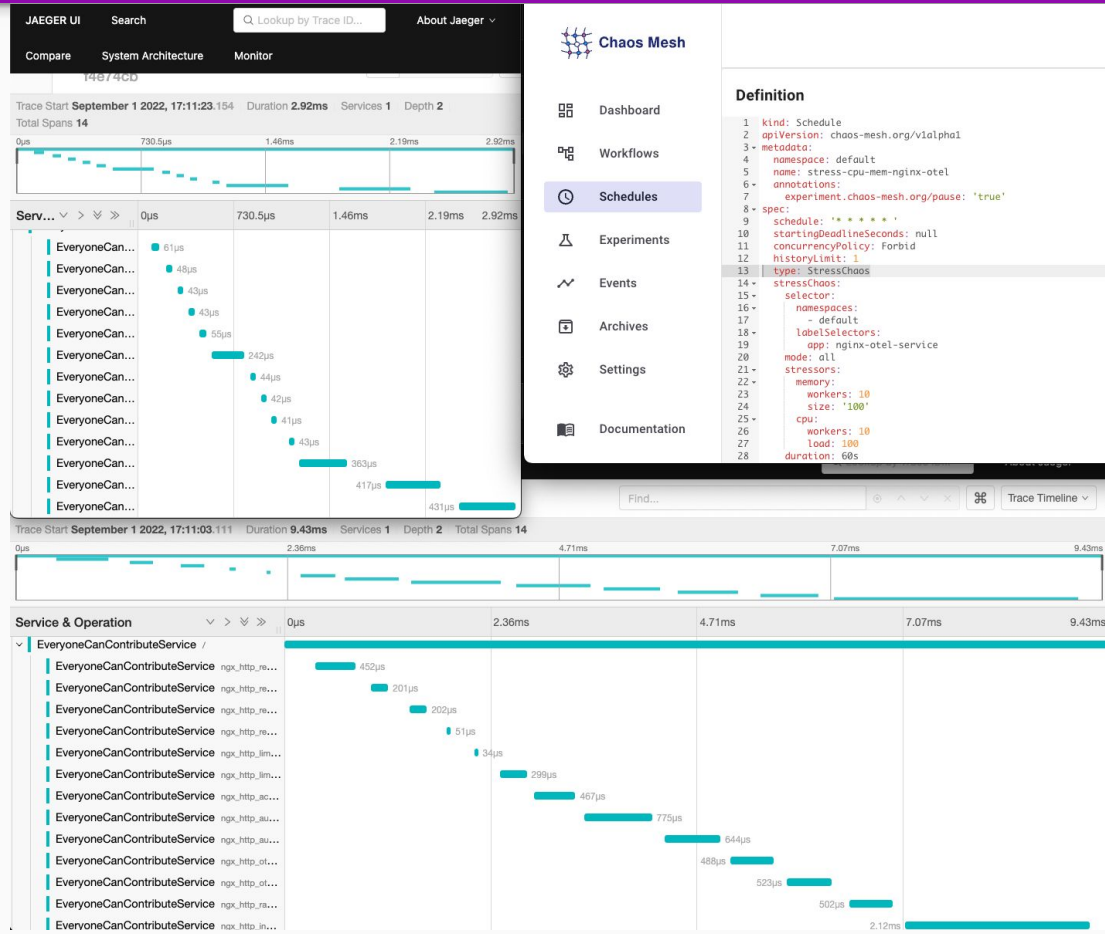
@dnsmichi

Chaos Mesh

- Stress test CPU & memory
- LabelSelector
  - app: nginx-otel-service

Jaeger

- Trace time++



https://gitlab.com/everyonecancontribute/observability/k8s-o11y-chaos/-/blob/main/kubernetes/chaos/stress-cpu-mem-nginx-otel.yml

Ideas

@dnsmichi

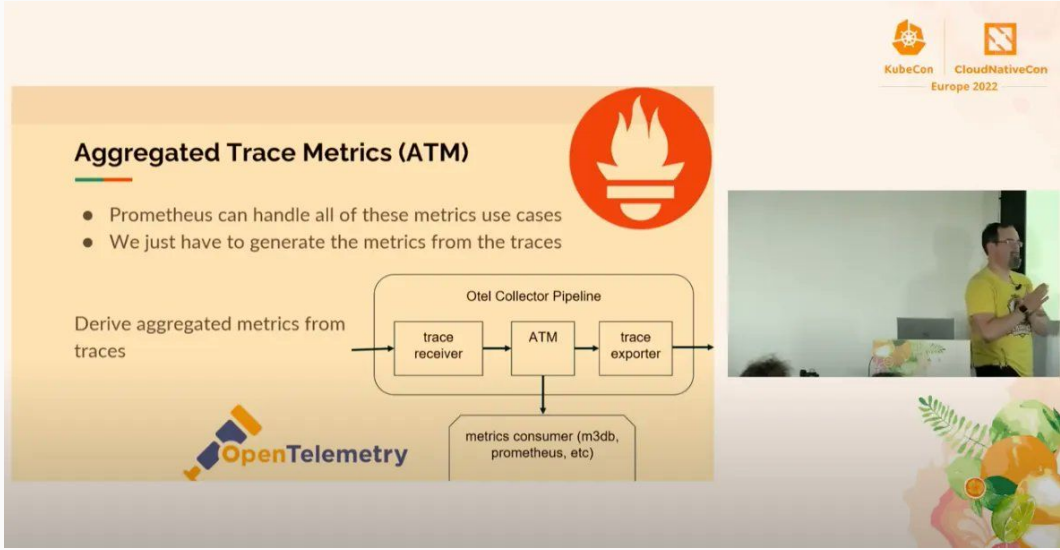
Exemplars

Link metrics with traces

Aggregated trace metrics

Create metrics from traces

Alerts and Chaos



Traces for Kubernetes System Components

Prometheus and Jaeger: A Match Made in Heaven! https://www.youtube.com/watch?v=EvzTFUMqCXY
Jaeger; Present and future, with Aggregated trace metrics: https://www.youtube.com/watch?v=NWQD_uCGP6k
https://kubernetes.io/docs/concepts/cluster-administration/system-traces/

Beyond Observability

Photo by Michael Friedrich

OSAD
OPEN SOURCE AUTOMATION DAYS

@dnsmichi

eBPF

@dnsmichi

"Instead of relying on static counters and gauges exposed by the operating system, **eBPF enables the collection & in-kernel aggregation of custom metrics and generation of visibility events** based on a wide range of possible sources."

Cilium, network connectivity security and observability (Tetragon)

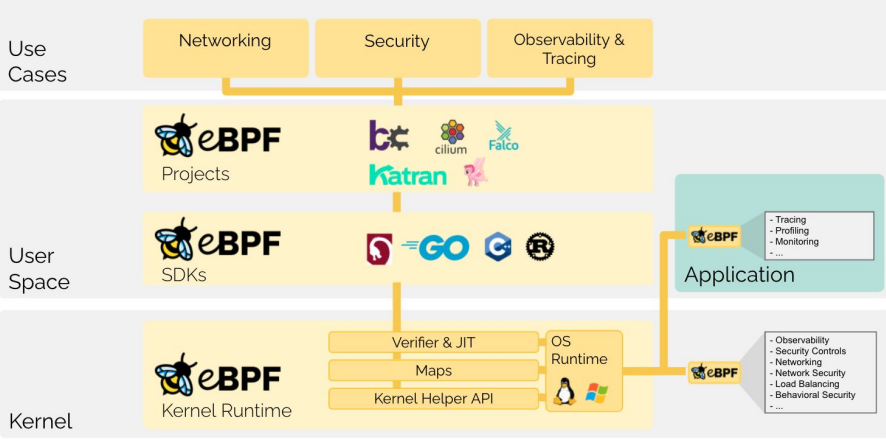
BumbleBee, build, run and distribute eBPF programs using OCI images

Pixie, Kubernetes observability for developers, auto-instrumented, scriptable

Falco, Kubernetes threat detection engine

Parca, Continuous Profiling

Tracee, Runtime Security and Forensics



https://ebpf.io/ - Image credit https://o11y.love/topics/collections-specs/#ebpf

Observability with eBPF

@dnsmichi

Where does eBPF fit into Kubernetes Observability?

Does it complement / integrate with Prometheus metrics?

eBPF, SLOs, alerts and Chaos?

Cilium Tetragon open-sourced at KubeCon EU

Future challenges and talks, yay! 💡

```
mfriedrich@Michaels-MacBook-Pro-2:~/dev/everyonecancontribute/observability/cpp-dns-leaker/kube-prometheus
main ● ●   -zsh   64%   12 GB
🚀 process monitoring/alertmanager-main-0 /bin/alertmanager --config.file=/etc/alertmanager/config/alertmanager.yaml --storage.path=/alertmanager --data.ret
ention=120h --cluster.listen-address=[10.42.2.7]:9094 --web.listen-address=:9093 --web.route-prefix=/ --cluster.peer=alertmanager-main-0.alertmanager-operat
ed:9094 --cluster.peer=alertmanager-main-1.alertmanager-operated:9094 --cluster.peer=alertmanager-main-2.alertmanager-operated:9094 --cluster.reconnect-time
out=5m
🚀 process monitoring/alertmanager-main-0 /bin/prometheus-config-reloader --listen-address=:8080 --reload-url=http://localhost:9093/-/reload --watched-dir=/
etc/alertmanager/config
🚀 process monitoring/prometheus-k8s-0 /bin/prometheus-config-reloader --listen-address=:8080 --reload-url=http://localhost:9090/-/reload --config-file=/etc
/prometheus/config/prometheus.yaml.gz --config-envsubst-file=/etc/prometheus/config_out/prometheus.env.yaml --watched-dir=/etc/prometheus/rules/prometheus-k
8s-rulefiles-0
🚀 process monitoring/prometheus-k8s-0 /bin/prometheus --web.console.templates=/etc/prometheus/consoles --web.console.libraries=/etc/prometheus/console_libr
aries --storage.tsdb.retention.time=24h --config.file=/etc/prometheus/config_out/prometheus.env.yaml --storage.tsdb.path=/prometheus --web.enable-lifecycle
--web.route-prefix=/ --web.config.file=/etc/prometheus/web_config/web-config.yaml
🚀 process default/tiefighter /usr/bin/netserver -D
🚀 process default/tiefighter /bin/bash
🚀 process default/tiefighter /usr/bin/whoami
💥 exit    default/tiefighter /usr/bin/whoami  0
🚀 process default/tiefighter /bin/cat /etc/os-release
💥 exit    default/tiefighter /bin/cat /etc/os-release 0
🚀 process default/tiefighter /sbin/apk update
💥 exit    default/tiefighter /sbin/apk update 0
🚀 process default/tiefighter /sbin/apk add curl
💥 exit    default/tiefighter /sbin/apk add curl 0
🚀 process default/tiefighter /usr/bin/curl everyonecancontribute.com
💥 exit    default/tiefighter /usr/bin/curl everyonecancontribute.com 0
🚀 process default/tiefighter /usr/bin/curl https://everyonecancontribute.com
💥 exit    default/tiefighter /usr/bin/curl https://everyonecancontribute.com 60
🚀 process default/cpp-dns-leaker-service-o11y-55d9498b7f-6w86z /usr/local/bin/cpp-dns-leaker
🔌 close   monitoring/alertmanager-main-0 /bin/alertmanager tcp 10.42.2.7:34083 -> 10.42.2.1:33550
🔌 close   monitoring/prometheus-adapter-7bf7ff5b67-h89nh /adapter tcp 10.42.2.6:11033 -> 10.42.2.1:54360
🔌 close   monitoring/prometheus-adapter-7bf7ff5b67-h89nh /adapter tcp 10.42.2.6:11033 -> 10.42.2.1:54362
📤 sendmsg monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:738 -> 192.168.1.5:10250 bytes 40
📤 sendmsg monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:738 -> 192.168.1.5:10250 bytes 35
📤 sendmsg monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:738 -> 192.168.1.5:10250 bytes 35
📤 sendmsg monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:738 -> 192.168.1.5:10250 bytes 35
📤 sendmsg monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:738 -> 192.168.1.5:10250 bytes 35
📤 sendmsg monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:33315 -> 10.42.2.1:50156 bytes 157
📤 sendmsg monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:33315 -> 10.42.2.1:50154 bytes 159
🔌 close   monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:33315 -> 10.42.2.1:50156
🔌 close   monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:33315 -> 10.42.2.1:50154
📤 sendmsg monitoring/prometheus-k8s-0 /bin/prometheus tcp 10.42.2.8:14568 -> 10.42.2.6:6443 bytes 40
```

https://www.youtube.com/watch?v=wv0AhY5-cJU

Hacking Kubernetes

@dnsmichi

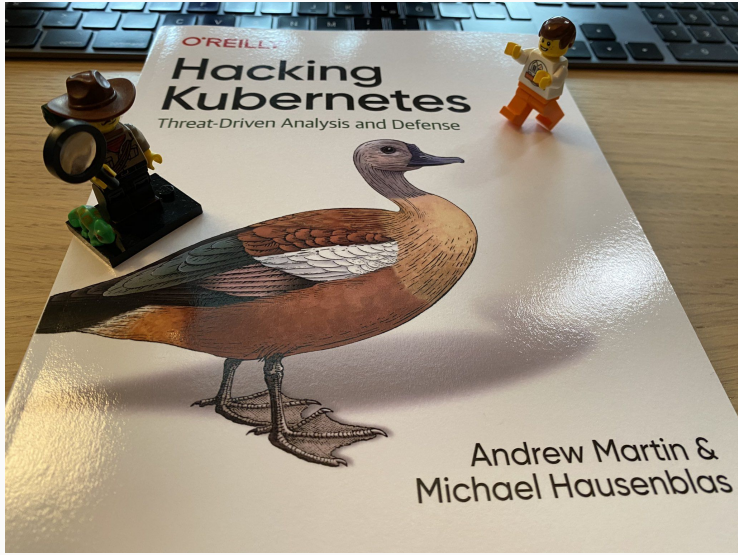
Security Defaults in Kubernetes need improvements

Impersonate the attacker

Use Chaos Engineering principles for pen testing?

Create a Chaos Experiment that downloads and installs malicious software

Observability for better Security?



https://www.oreilly.com/library/view/hacking-kubernetes/9781492081722/

Photo by Michael Friedrich

Confidence, Finally

Photo by Michael Friedrich

OSAD
OPEN SOURCE AUTOMATION DAYS

@dnsmichi

Chaos for SRE, Dev, Ops, DevOps, DevSecOps

@dnsmichi

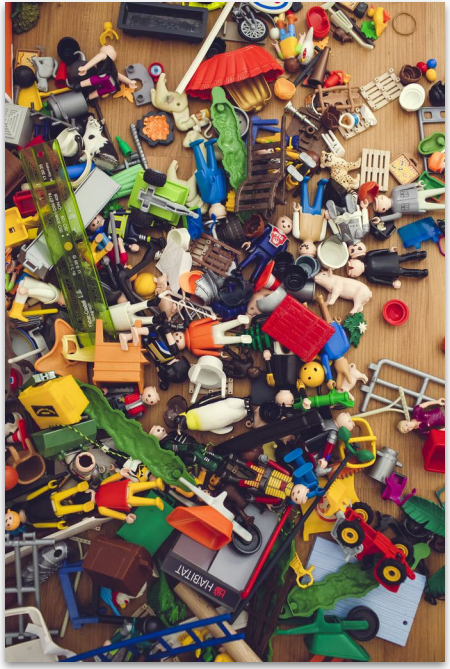
SRE: CPU overload, HTTP requests blocked (golden signals)

Dev: Many API clients not closing the connections

Ops: Intercept DNS traffic, do not resolve

DevOps: Registry proxy limiting container pulls

DevSecOps: Break security policies, download software



https://chaos-mesh.org/docs/simulate-dns-chaos-on-kubernetes/
https://chaos-mesh.org/docs/developer-guide-overview/

Photo by Markus Spiske on Unsplash

Your own Chaos: Iterate

- Know the limits of Chaos
  - Avoid Chaos Inception
  - Do not harm existing workflows or teams
  - Staging envs to prevent data loss
- Chaos Engineering does not solve all reliability issues
  - It can help with new perspectives 🌱

https://litmuschaos.github.io/litmus/experiments/categories/contents/
https://sharpend.io/the-limitations-of-chaos-engineering/

Photo by Diego PH on Unsplash

Continuous Observability with Chaos

- CI/CD Environments in Kubernetes
  - Observability and Chaos OOTB
  - Feedback in Merge Requests
  - Alerts & Incidents for everyone
- Continuous Delivery and Observability
  - Run Chaos Experiments in production
  - “Red team” for Observability tests

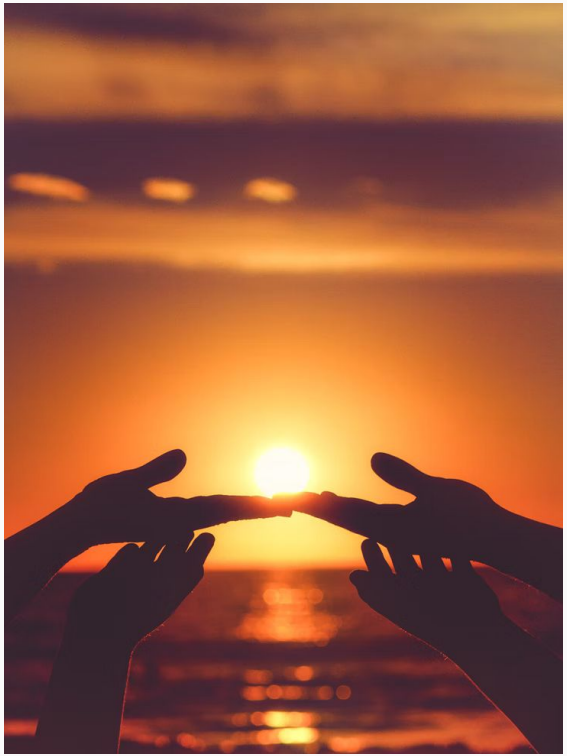


Photo by Alonso Reyes on Unsplash

Confidence with Chaos

Bring chaos into Observability

- Chaos workflows
- Verify alerts and SLOs

Iterate and innovate

MLOps and Chaos next?

So many great chaos engineering tools
https://o11y.love/topics/chaos-engineering/

Photo by Pavel Neznanov on Unsplash

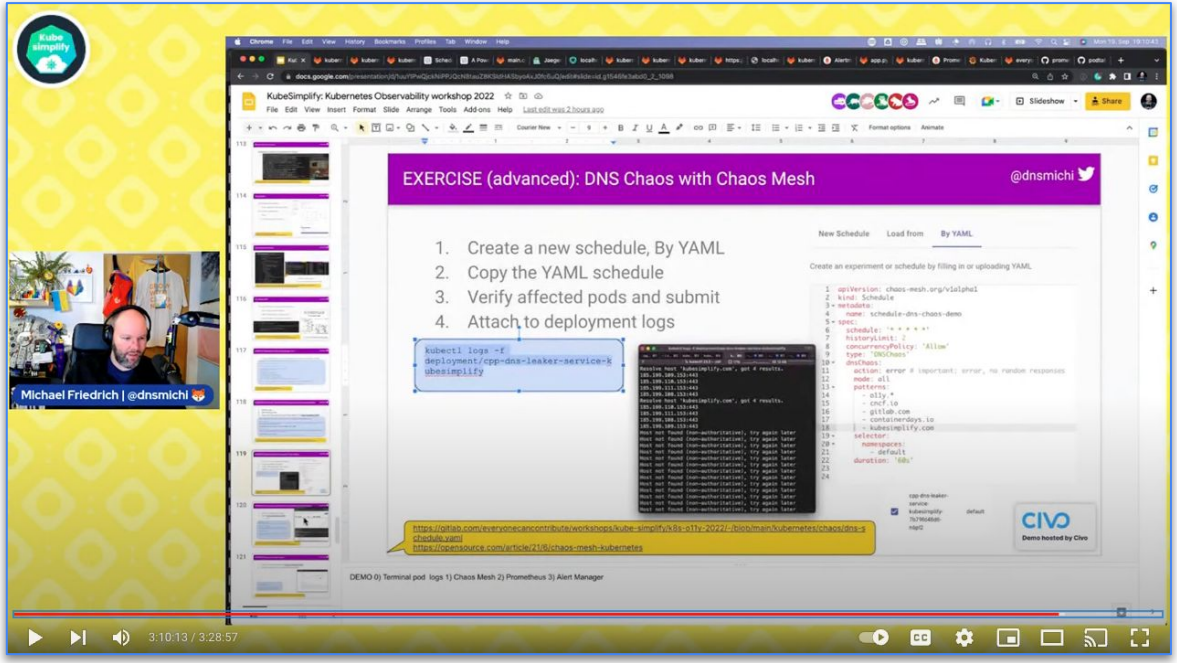
More Observability?

Practice in my **free Kubernetes Observability** workshop for Kube Simplify: https://o11y.love/learning-resources/#workshops

Learn more: https://o11y.love/ 💡

Subscribe to my newsletter: https://opsindev.news/ 📈

Slides available at https://dnsmichi.at/talks/ 🌱



OSAD

OPEN SOURCE AUTOMATION DAYS

@dnsmichi

Confidence with Chaos for your Kubernetes Observability

Michael Friedrich
Senior Developer Evangelist at GitLab

@dnsmichi

EveryoneCanContribute.com

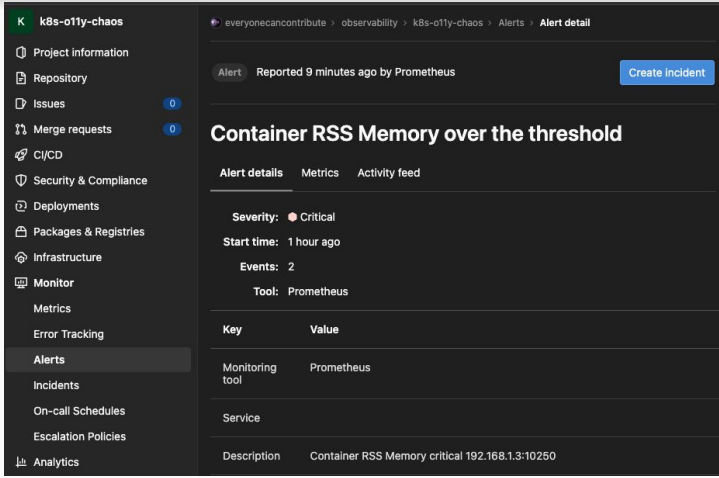
OSAD
OPEN SOURCE AUTOMATION DAYS

Demo case: Alerts in GitLab

@dnsmichi

HTTP endpoint

webhook_configs in Alert Manager




```
local kp =
  (import 'kube-prometheus/main.libsonnet') +
  {
    values+:: {
      alertmanager+: {
        config: importstr 'alertmanager-config.yaml',
      },
    },
  };
```

```
- name: 'gitlab'
  webhook_configs:
  - http_config:
      bearer_token: 'xxx'
    url:
'https://gitlab.com/everyonecancontribute/observability/k8s-o11y-chaos/al
erts/notify/k8s-prometheus/xxx.json'
    send_resolved: true


  - match:
      alertname: Watchdog
    receiver: 'null'
  - continue: true
    receiver: gitlab
```

https://gitlab.com/everyonecancontribute/observability/k8s-o11y-chaos/-/alert_management

Chaos Mesh on Civo Cloud

```
helm repo add chaos-mesh https://charts.chaos-mesh.org

helm search repo chaos-mesh

kubectl create ns chaos-testing

helm upgrade chaos-mesh chaos-mesh/chaos-mesh \
--namespace=chaos-testing \
--version 2.2.0 \
--set dnsServer.create=true \
--set dashboard.create=true \
--set dashboard.securityMode=false \
--set chaosDaemon.hostNetwork=true \
--set chaosDaemon.runtime=containerd \
--set chaosDaemon.socketPath=/run/k3s/containerd/containerd.sock

kubectl --namespace chaos-testing port-forward svc/chaos-dashboard 2333
```

https://chaos-mesh.org/docs/production-installation-using-helm/
Follow https://github.com/civo/kubernetes-marketplace/blob/master/chaos-mesh/install.sh
to apply Civo specific installation parameters for k3s.