

Are you testing your Observability?

GoDays 2020, Berlin





Thanos



Red Hat



What is observability 

Why Prometheus? 

Signals 

Metrics are essential ❄️

Prometheus - How to expose? ❄️

This is a title

```
// Register standard Go metric collectors, which are by default registered when using global registry.
reg.MustRegister(
    prometheus.NewGoCollector(),
    prometheus.NewProcessCollector(prometheus.ProcessCollectorOpts{}),
    prometheus.NewGaugeFunc(prometheus.GaugeOpts{
        Namespace: namespace,
        Name:       "configured_max_objects",
        Help:       "Configured maximum number of objects cache can take.",
    }, func() float64 { return float64(*maxObjectsCount) }),
)
```

and this a subtitle

```
mux.Handle("/metrics", promhttp.HandlerFor(reg, promhttp.HandlerOpts{}))  
mux.Handle("/set", http.HandlerFunc(c.SetHandler))  
mux.Handle("/get", http.HandlerFunc(c.GetHandler))
```

get

Holy battle; pull vs push

Instrumenting Go applications

Common pitfalls

Globals

Intro: define metric; register; expose HTTP; usage

Problems: terrible for libs

Solution: custom registered

Counter vs Gauges

Problems ??

Queue size vs added/removed?

Gauge, reset etc loss of info

Histogram vs Summaries

Yolo histogram buckets

Not SLO based

Random string in labels

No Testing

Hardcoding namespace in libraries/package

WrapRegisterer

Not using middlewares

Solution:

Reuse internally

Convention/reuse dashboards; mixins!

~~Q/A~~

Discussion

Reference:

-