



Best Practices for Debugging and Monitoring on Kubernetes

Thomas Rampelberg



Honest Status Page

@honest_update

Follow



We replaced our monolith with micro services so that every outage could be more like a murder mystery.

4:10 PM - 7 Oct 2015

 Hello GoSF!

Thomas Rampelberg

Software Engineer @ Buoyant

 @grampelberg

 grampelberg



What problem needs to be solved?



Alerting

- Downtime
- Exceptions
- Service Level Agreements



Forensics

- Memory leaks
- Performance issues
- Environment specific failures

Monitoring for Alerting



Downtime

- External monitoring
- livenessProbe/readinessProbe
- kube-state-metrics



Pingdom:

<https://www.pingdom.com/>

Configuring livenessProbe:

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/>

Kube-state-metrics:

<https://github.com/helm/charts/tree/master/stable/kube-state-metrics>



Exceptions

```
func readIntoBytes(filename string) ([]byte, error) {  
    file, err := static.Templates.Open(filename)  
    if err != nil {  
        return nil, err  
    }  
    defer file.Close()  
  
    buf := new(bytes.Buffer)  
    buf.ReadFrom(file)  
  
    return buf.Bytes(), nil  
}
```

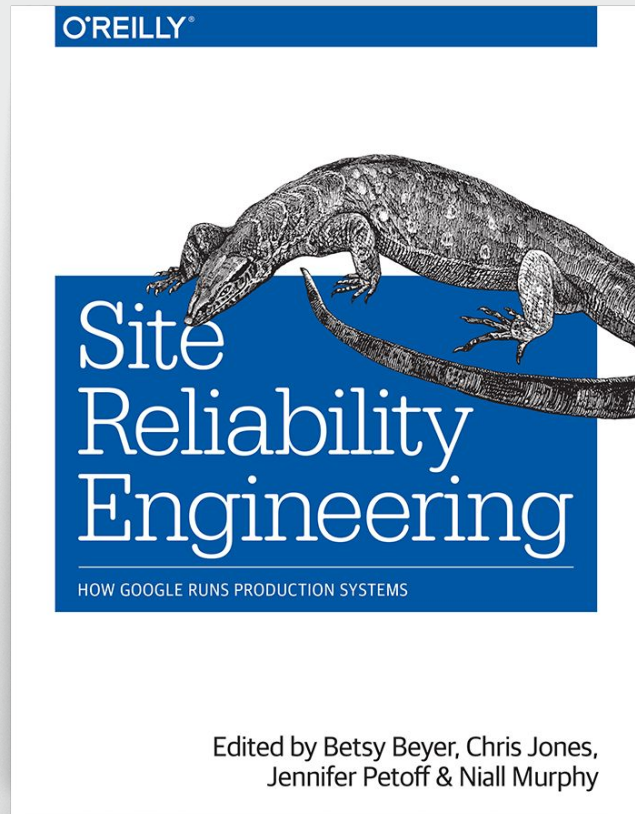




Golden Signals

- Latency
- Traffic
- Errors
- Saturation

<http://bit.ly/golden-signals>





Every request matters

- Tail latency is important
- Users see responses
- Latency is not normally distributed

Site	# of requests	page loads that would experience the 99%'lie [(1 - (.99 ^ N)) * 100%]
amazon.com	190	85.2%
kohls.com	204	87.1%
jcrew.com	112	67.6%
saksfifthavenue.com	109	66.5%
--	--	--
nytimes.com	173	82.4%
cnn.com	279	93.9%
--	--	--
twitter.com	87	58.3%
pinterest.com	84	57.0%
facebook.com	178	83.3%
--	--	--
google.com (yes, that simple noise-free page)	31	26.7%
google.com search for "http requests per page"	76	53.4%

Monitoring for Forensics



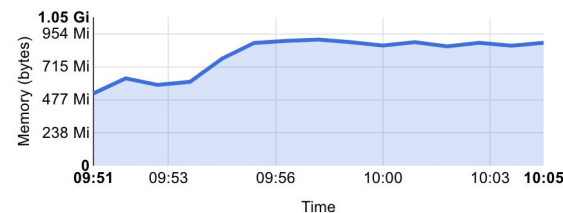
Kubernetes

- CPU
- Memory

CPU usage



Memory usage ⓘ



☀ Prometheus

- Kubernetes Standard
- Time Series
- Alerts
- Extra Metrics



Install: <https://github.com/helm/charts/tree/master/stable/prometheus>
Tracked metrics: https://github.com/prometheus/node_exporter#enabled-by-default

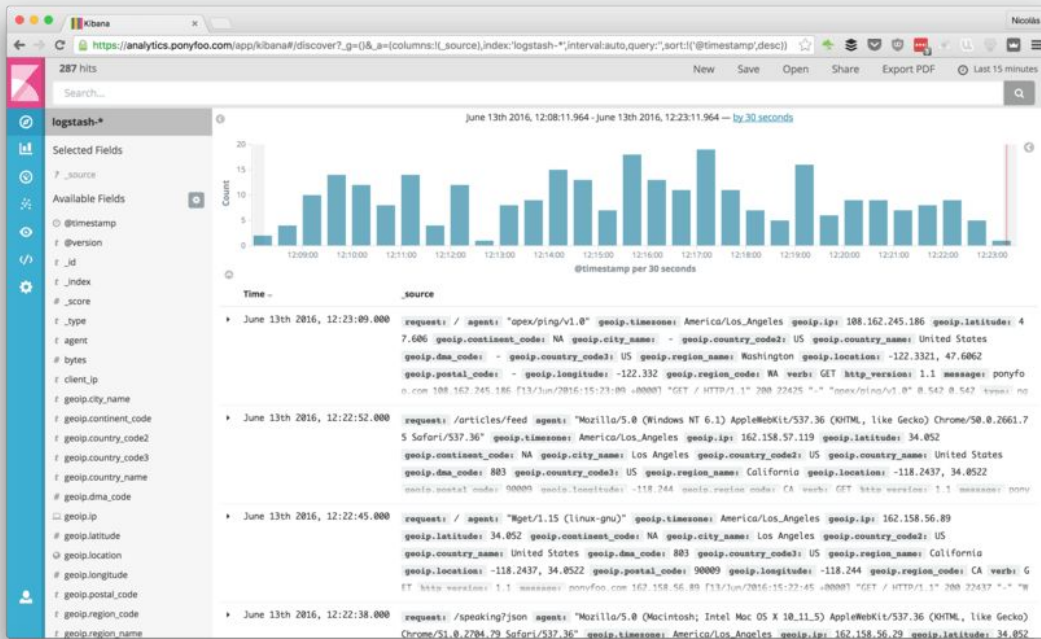
Application Instrumentation

```
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 1.9864e-05
go_gc_duration_seconds{quantile="0.25"} 5.2299e-05
go_gc_duration_seconds{quantile="0.5"} 7.1593e-05
go_gc_duration_seconds{quantile="0.75"} 0.000151335
go_gc_duration_seconds{quantile="1"} 0.003560545
go_gc_duration_seconds_sum 0.058749902
go_gc_duration_seconds_count 254
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 59
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.10.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 6.619704e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.465679856e+09
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.50257e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 6.977458e+06
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time that the garbage collector has been using since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 3.924477433282826e-05
```

```
func main() {
    recordMetrics()

    http.Handle("/metrics", promhttp.Handler())
    http.ListenAndServe(":2112", nil)
}
```

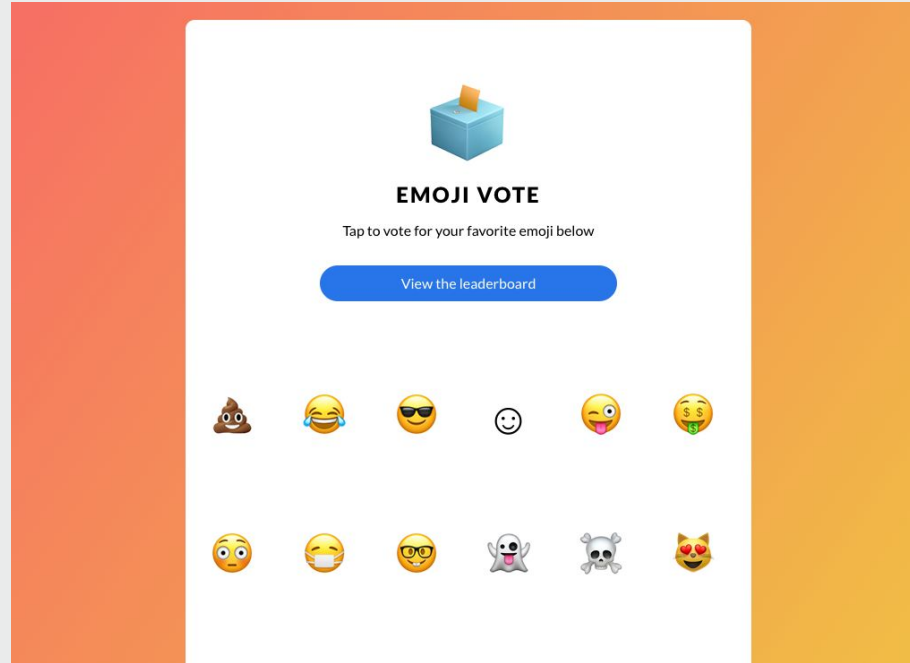
Log Aggregation



Howto: <https://kubernetes.io/docs/tasks/debug-application-cluster/logging-elasticsearch-kibana/>



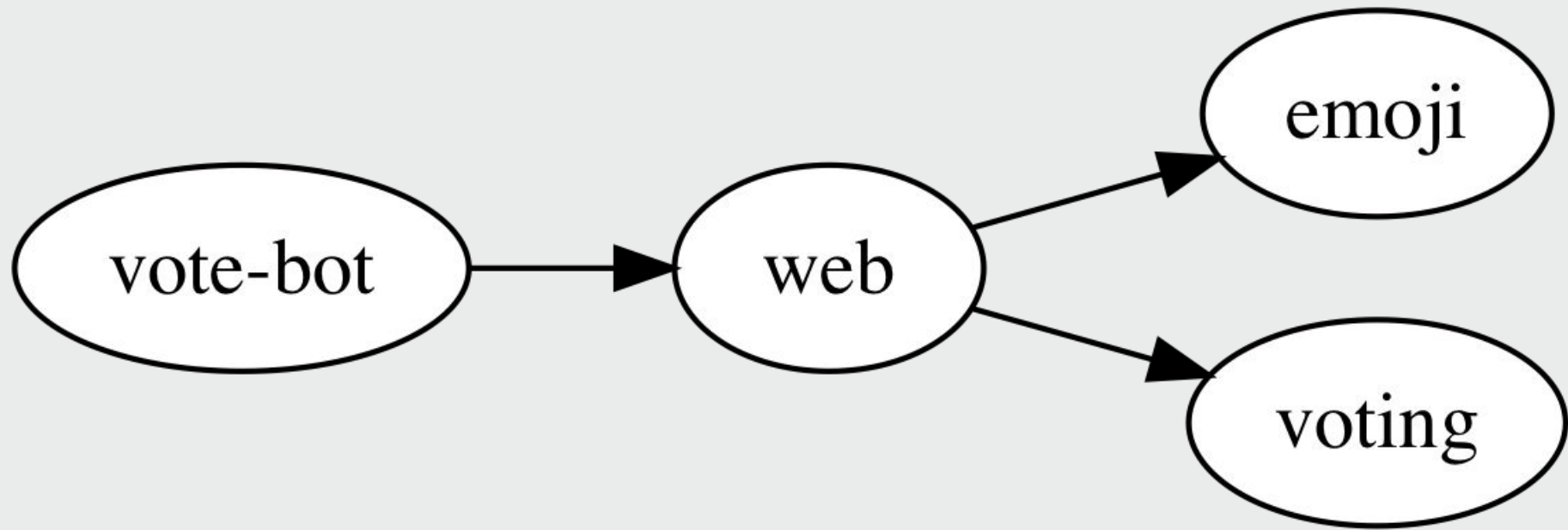
Get your votes in



<https://emoji.l5d.io>



The Architecture





The Problem

Uh oh.



For the sake of this demo, voting for 🐛
always returns an error.

Get your mind out of the gutter, and [pick another!](#)

Select again



Check the state

- ☐ Service running?
- ☐ Health checks passing?
- ☐ Logs look okay?



 No problem here!

☒ Service running!

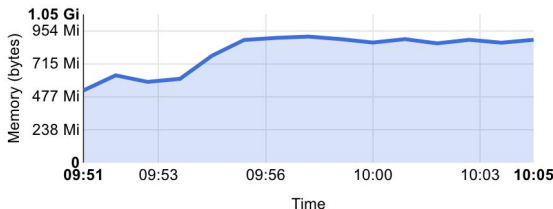
☐ Health checks passing?

☐ Logs look okay?

CPU usage



Memory usage ⓘ



Everything is fine

- ☒ Service running!
- ☒ Health checks passing!
- ☐ Logs look okay?

Deployments		
Name ▾	Labels	Pods
<input checked="" type="checkbox"/> vote-bot	app: vote-bot	1 / 1
<input checked="" type="checkbox"/> web	app: web-svc	1 / 1
<input checked="" type="checkbox"/> emoji	app: emoji-svc	1 / 1
<input checked="" type="checkbox"/> voting	app: voting-svc	1 / 1

 Okay, *something* is fishy



Service running!



Health checks passing!



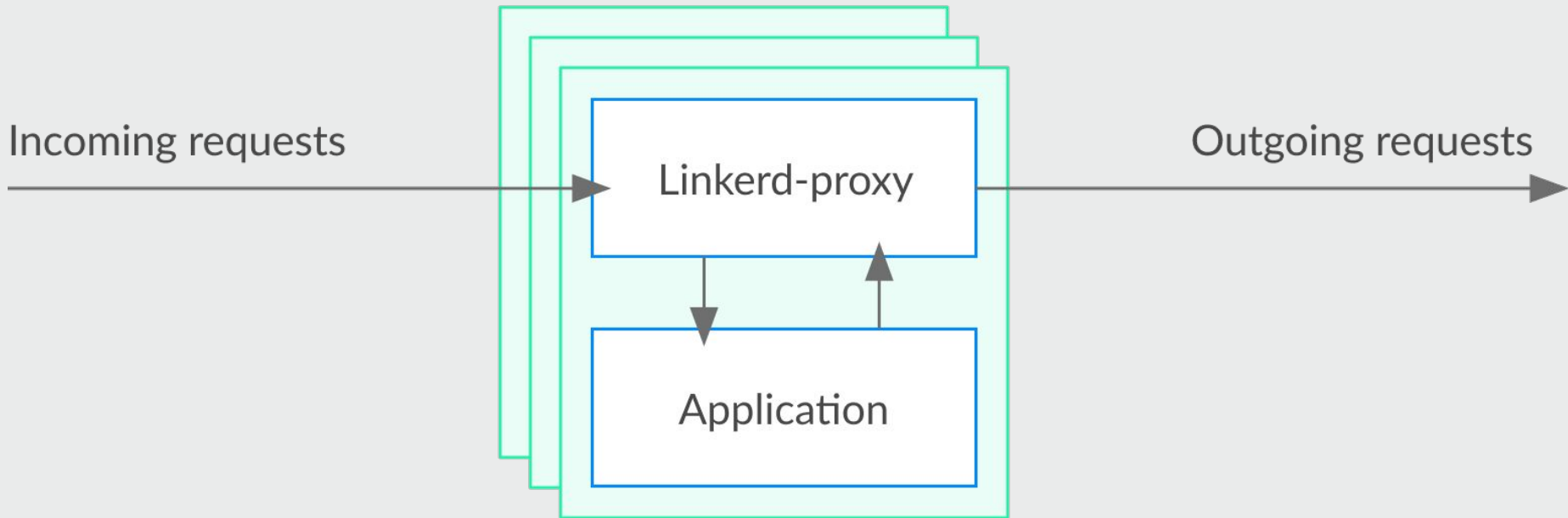
Logs look ... something?

```
2018/11/13 22:26:58 Error serving request [&{GET /api/vote?choice=:poop: HTTP/1.1 1 1 map[User-Agent:[Go-http-client/1.1] Accept-Encoding:[gzip]] {} <nil> 0 [] false web-svc.emojivoto:80 map[choice[:poop:]] map[] <nil> map[] 10.4.0.31:34560 /api/vote?choice=:poop: <nil> <nil> <nil> 0xc420481b80}): rpc error: code = Unknown desc = ERROR
2018/11/13 22:27:01 Error serving request [&{GET /api/vote?choice=:poop: HTTP/1.1 1 1 map[Accept-Encoding:[gzip] User-Agent:[Go-http-client/1.1]] {} <nil> 0 [] false web-svc.emojivoto:80 map[choice[:poop:]] map[] <nil> map[] 10.4.0.31:34590 /api/vote?choice=:poop: <nil> <nil> <nil> 0xc420481f00}): rpc error: code = Unknown desc = ERROR
2018/11/13 22:27:15 Error serving request [&{GET /api/vote?choice=:poop: HTTP/1.1 1 1 map[Accept-Encoding:[gzip] User-Agent:[Go-http-client/1.1]] {} <nil> 0 [] false web-svc.emojivoto:80 map[choice[:poop:]] map[] <nil> map[] 10.4.0.31:34618 /api/vote?choice=:poop: <nil> <nil> <nil> 0xc420480f80}): rpc error: code = Unknown desc = ERROR
2018/11/13 22:27:18 Error serving request [&{GET /api/vote?choice=:poop: HTTP/1.1 1 1 map[User-Agent:[Go-http-client/1.1] Accept-Encoding:[gzip]] {} <nil> 0 [] false web-svc.emojivoto:80 map[choice[:poop:]] map[] <nil> map[] 10.4.0.31:34730 /api/vote?choice=:poop: <nil> <nil> <nil> 0xc420481380}): rpc error: code = Unknown desc = ERROR
2018/11/13 22:27:19 Error serving request [&{GET /api/vote?choice=:poop: HTTP/1.1 1 1 map[User-Agent:[Go-http-client/1.1] Accept-Encoding:[gzip]] {} <nil> 0 [] false web-svc.emojivoto:80 map[choice[:poop:]] map[] <nil> map[] 10.4.0.31:34754 /api/vote?choice=:poop: <nil> <nil> <nil> 0xc420061280}): rpc error: code = Unknown desc = ERROR
2018/11/13 22:27:24 Error serving request [&{GET /api/vote?choice=:poop: HTTP/1.1 1 1 map[User-Agent:[Go-http-client/1.1] Accept-Encoding:[gzip]] {} <nil> 0 [] false web-svc.emojivoto:80 map[choice[:poop:]] map[] <nil> map[] 10.4.0.31:34766 /api/vote?choice=:poop: <nil> <nil> <nil> 0xc420061980}): rpc error: code = Unknown desc = ERROR
2018/11/13 22:27:33 Error serving request [&{GET /api/vote?choice=:poop: HTTP/1.1 1 1 map[User-Agent:[Go-http-client/1.1] Accept-Encoding:[gzip]] {} <nil> 0 [] false web-svc.emojivoto:80 map[choice[:poop:]] map[] <nil> map[] 10.4.0.31:34798 /api/vote?choice=:poop: <nil> <nil> <nil> 0xc420060040}): rpc error: code = Unknown desc = ERROR
2018/11/13 22:27:38 Error serving request [&{GET /api/vote?choice=:poop: HTTP/1.1 1 1 map[User-Agent:[Go-http-client/1.1] Accept-Encoding:[gzip]] {} <nil> 0 [] false web-svc.emojivoto:80 map[choice[:poop:]] map[] <nil> map[] 10.4.0.31:34864 /api/vote?choice=:poop: <nil> <nil> <nil> 0xc420060880}): rpc error: code = Unknown desc = ERROR
```

Linkerd: Microservice Forensics

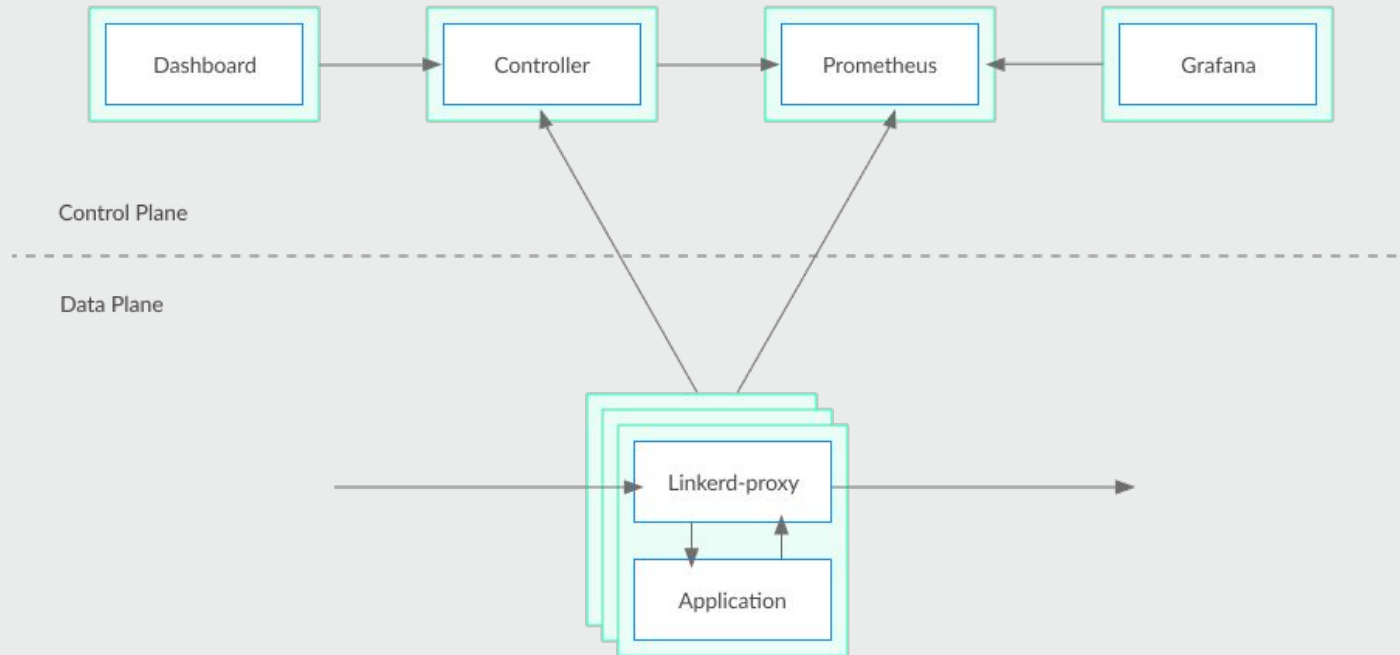


What is Linkerd?



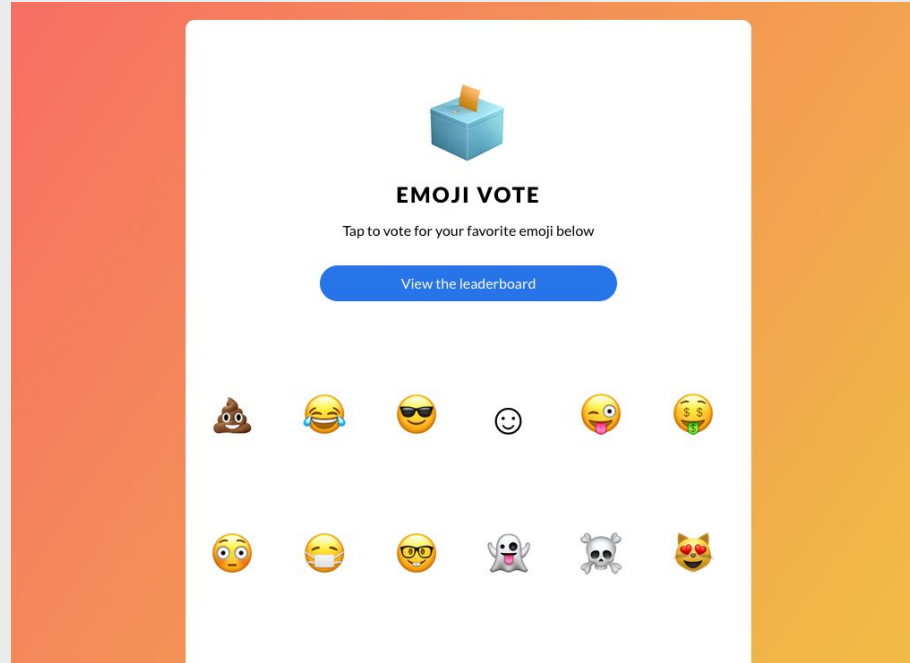


Architecture





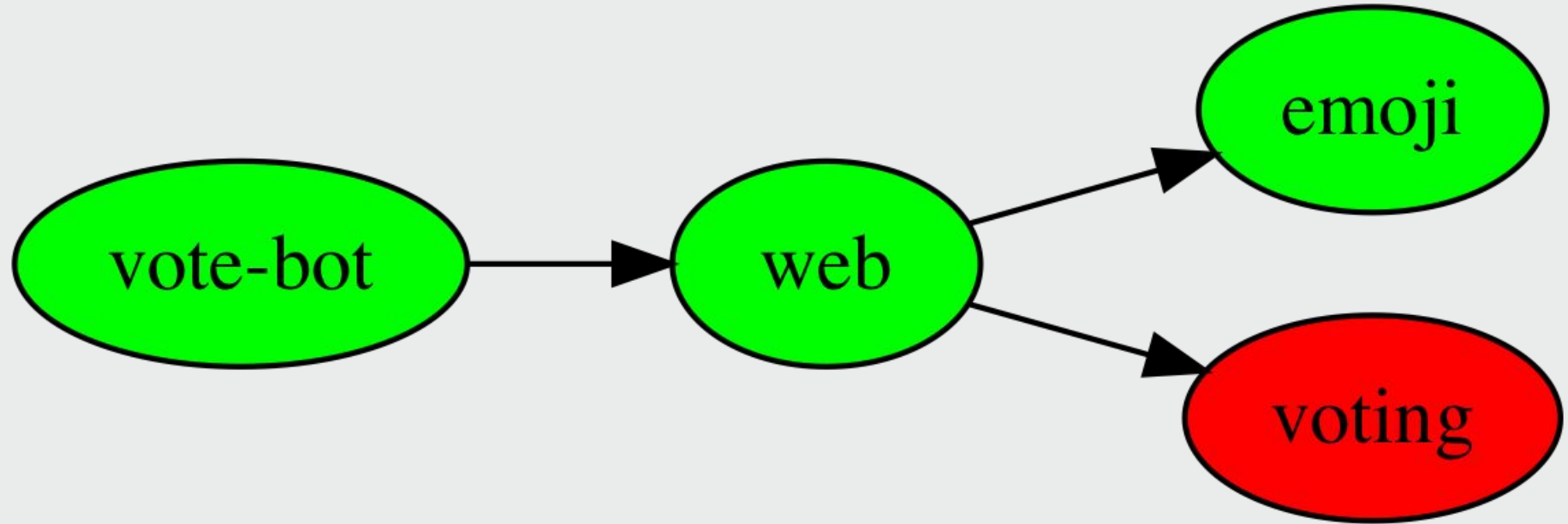
Get your votes in



<https://emoji.l5d.io>



The Culprit





What is possible?

- Alerting
- Anomaly Detection
- Autoscaling
- Capacity Planning
- Chargeback
- Intelligent Deployments
- SLA Compliance



LINKERD

Slides

Tutorial

<http://bit.ly/linkerd-tutorial>

Get Started!

<https://bit.ly/linkerd-get-started>
