Kubernetes Community Days

Top 10 Metrics to Observe in Kubernetes (including stuffs around K8s)

Steve Ng
Developer Relations Lead
New Relic APJ

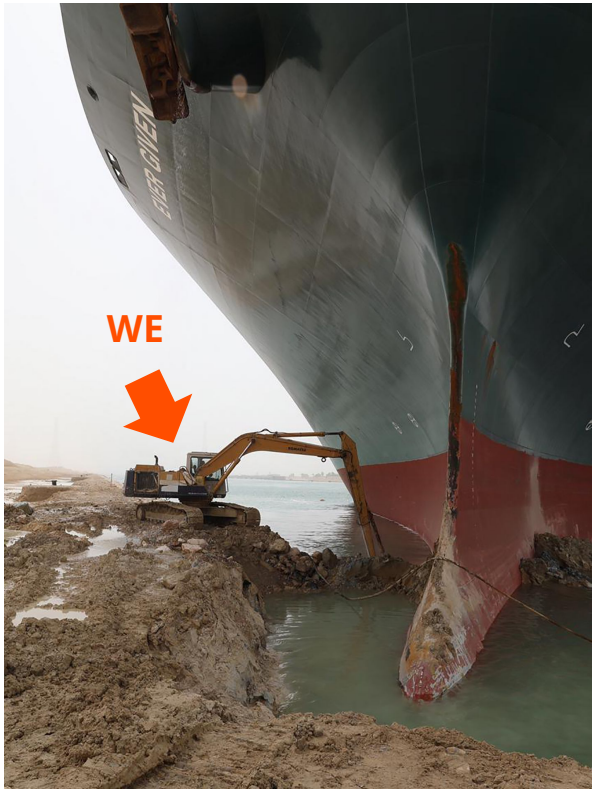
Let's Talk Kubernetes

Kubernetes Community Days

Kubernetes is awesome, and scary.

Sounds Familiar?

Kubernetes Community Days



Where to get Telemetry …

Kubernetes Community Days

cAdvisor

Metric Server

API Server

Node Exporter

Kube State Metrics

Observe Symptoms, not Problems

Opinion - if only I can pick **10** ~

Problem - Misconfiguration Oops!

Kubernetes Community Days

Why should you care?

No magic solution to fix fat fingers, incorrect characters, missing requirements, invaild inputs, etc ...

K8s - A combination of source control + deployment pipeline with **smoke tests** for each stage (Launchdarkly).

Leverage existing tools such as Infrastructure as Code, and Configuration Management for config consistency.

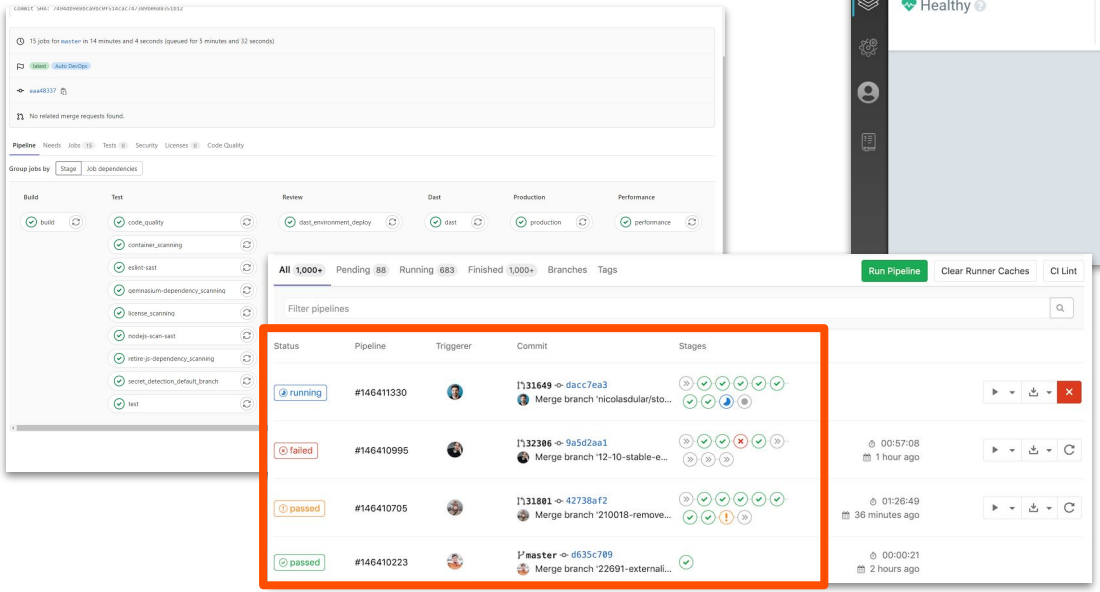
Tradeoff - increased cost (environments to test) & good Git practice, but still better compared to Production outages.

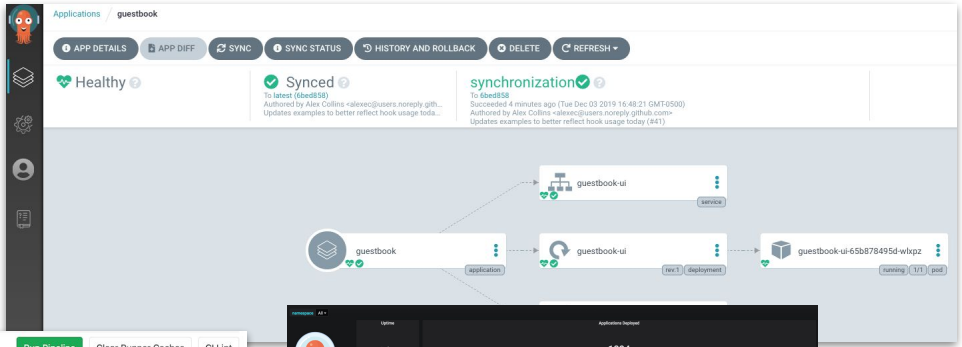
Possible solution in 2022 - **GitOps with CD** aka CNCF ArgoCD.

Misconfigurations

Problem - Misconfiguration Oops!

Where to find them?





Important > **Pipeline** [Status] (via **GitLab**)
[Analytics] coming based on DORA DevOps Research

[ArgoCD] is promising with GitOps
Guides for [Operators], and [Metrics] available

Problem - I need MOORRREEEE!

Kubernetes Community Days

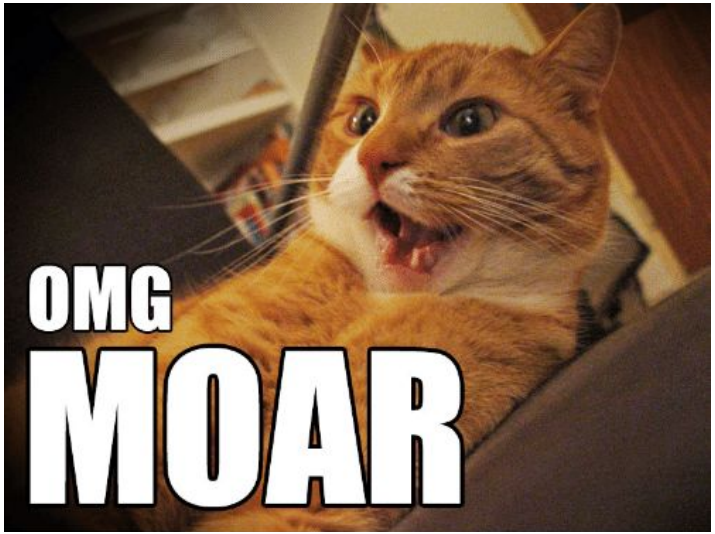
Why should you care?

Every system has a **DEFINITE** limit!
CPU limits - Throttling, Memory limits - OOMKilled.

The Debate of the Century - **set** quota or **no** quota!
Even **Google** recommends to set requests & limits!

Capacity planning is a lost art! How we plan,

- 30% free capacity - autoscale if needed.
- N+2 failover for services.
- Trend analysis - weekly, monthly & quarterly.
- Histogram heatmaps - anticipate spike.
- Forecast for features testing, and rollouts.



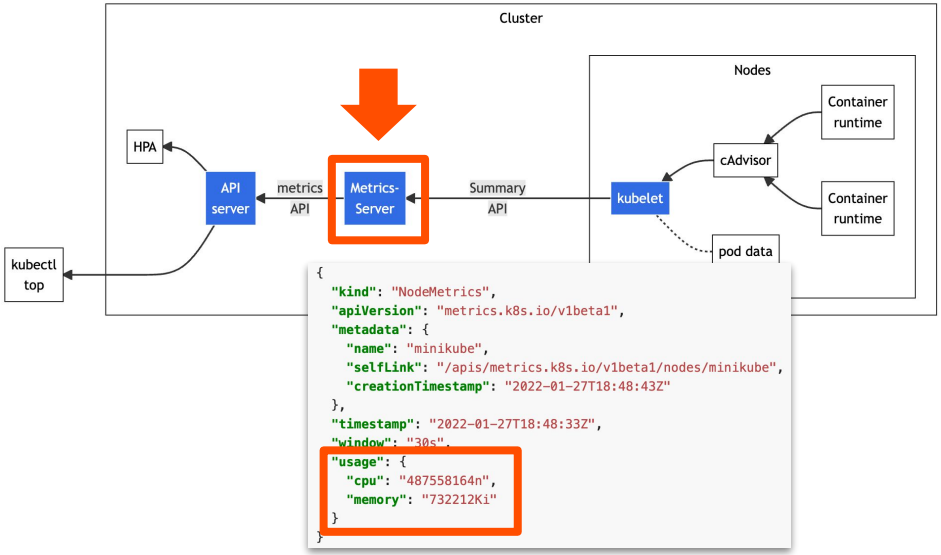
Give me MOAR, MOAR, MOAR!

Problem - I need MOORRREEEE!

Kubernetes Community Days

Where to find them?

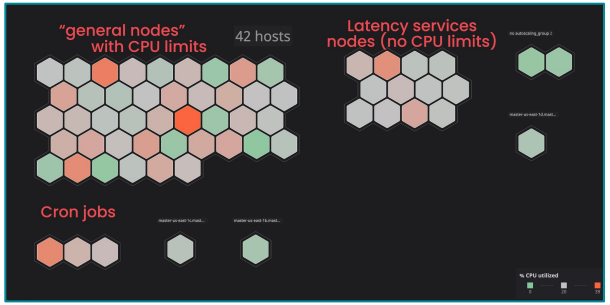
kubectl get --raw "XXX" | jq '.'

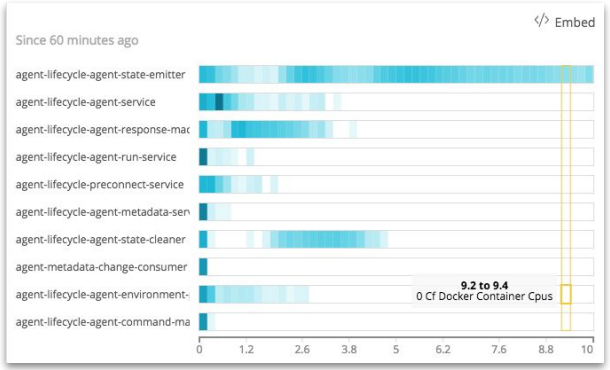


Important > CPU, Important > Memory

Metric Pipelines are important for Horizontal & Vertical Scaling

Idea - Isolating "No CPU Limits" services

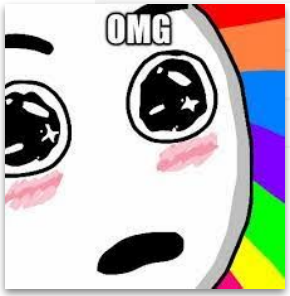




Anticipate spikes with heatmaps

Capacity Planning Made EASY

Kubernetes Community Days



https://learnk8s.io/kubernetes-instance-calculator

Problem - Network Headaches

Kubernetes Community Days

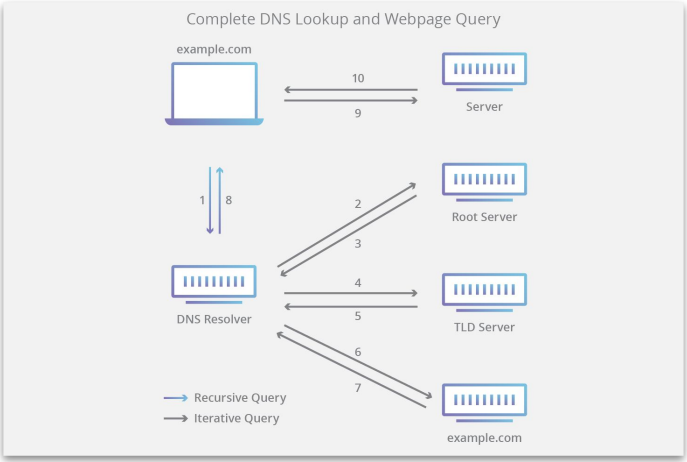
Why should you care?

Troubleshooting network issue is **NO FUN AT ALL** :(

K8s **v1.12** - **CoreDNS** is the recommended DNS Server, replacing **kube-dns**. K8s DNS schedules a DNS Pod and Service - tell containers to resolve DNS names.

Most common problems with DNS - **resolution errors**, or **network connectivity issues**. Will see **timeout** or **errors** with these problems.

Common problems with K8s **Ingress**, but remediation options are improving with providers.



What is a DNS Server?
CloudFlare

Problem - Network Headaches

Where to find them? - DNS

The Corefile configuration includes the following plugins of CoreDNS:

- errors: Errors are logged to stdout.
- health: Health of CoreDNS is reported to `http://localhost:8080/health`. In this extended syntax `lameduck` will make the process unhealthy then wait for 5 seconds before the process is shut down.
- ready: An HTTP endpoint on port 8181 will return 200 OK, when all plugins that are able to signal readiness have done so.
- kubernetes: CoreDNS will reply to DNS queries based on IP of the services and pods of Kubernetes. You can find more details about that plugin on the CoreDNS website. `ttl` allows you to set a custom TTL for responses. The default is 5 seconds. The minimum TTL allowed is 0 seconds, and the maximum is capped at 3600 seconds. Setting TTL to 0 will prevent records from being cached. The `pods insecure` option is provided for backward compatibility with *kube-dns*. You can use the `pods verified` option, which returns an A record only if there exists a pod in same namespace with matching IP. The `pods disabled` option can be used if you don't use pod records.
- prometheus: Metrics of CoreDNS are available at `http://localhost:9153/metrics` in Prometheus format (also known as OpenMetrics).
- forward: Any queries that are not within the cluster domain of Kubernetes will be forwarded to predefined resolvers (/etc/resolv.conf).
- cache: This enables a frontend cache.
- loop: Detects simple forwarding loops and halts the CoreDNS process if a loop is found.
- reload: Allows automatic reload of a changed Corefile. After you edit the ConfigMap configuration, allow two mi
changes to take effect.
- loadbalance: This is a round-robin DNS loadbalancer that randomizes the order of A, AAAA, and MX records in t

Kubernetes DNS (via CoreDNS)

Important > **Errors** (via CoreDNS)

```
2 errors like '^read udp .* i/o timeout$' occurred in last 30s
```

- `coredns_health_request_duration_seconds{}` - duration to process a HTTP query to the local `/health` endpoint. As this a local operation it should be fast. A (large) increase in this duration indicates the CoreDNS process is having trouble keeping up with its query load.
- `coredns_health_request_failures_total{}` - The number of times the internal health check loop failed to query `/health`.

Important > **Health** (via CoreDNS)

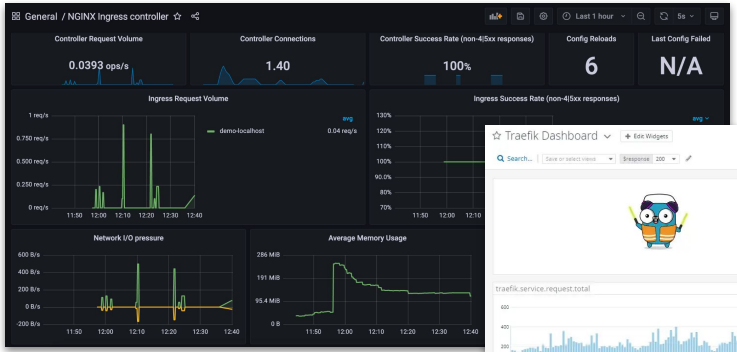
Verify CoreDNS is running well

```
.:53
2018/08/15 14:37:17 [INFO] CoreDNS-1.2.2
2018/08/15 14:37:17 [INFO] linux/amd64, go1.10.3, 2e322f6
CoreDNS-1.2.2
linux/amd64, go1.10.3, 2e322f6
2018/08/15 14:37:17 [INFO] plugin/reload: Running configuration MD5 = 24e6c59e83ce706f07bcc82c31b1ea1c
```

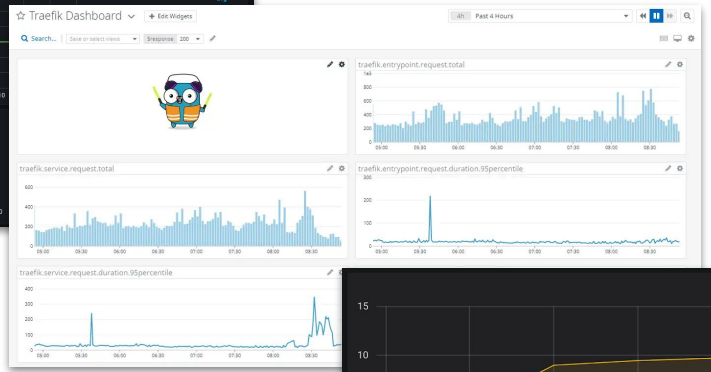
kubectl logs --namespace=kube-system -l k8s-app=kube-dns

Problem - Network Headaches

Where to find them? - Ingress

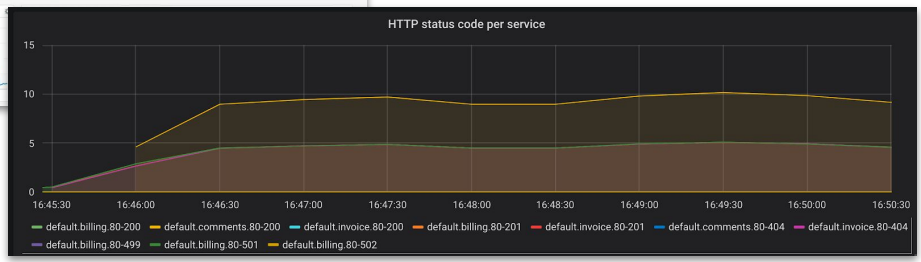


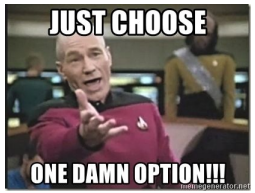
[NGINX]



[Traefik]

[Kong]





If I can only pick ONE,
Important > Error Rate (providers)

Problem - Help! My Pods are stuck!

Kubernetes Community Days

Why should you care?

Pretty common to see pods get stuck!

Issues such as insufficient resources, reached limits, incorrect image, misconfiguration, unable to resolve names, etc ... where should I start?

kubectl **describe** pod X is best to check what's going on.

K8s is getting better in errors, when things go wrong. Consider customizing your **termination message**.

Tip - Only push **Errors** logs into a centralized service!

Pods need help too

Problem - Help! My Pods are stuck!

Kubernetes Community Days

Where to find them?

Set up observers in **Kube State Metrics** - [pod-metric](#).

Important > **kube_pod_status_phase** - <Pending|Running|Succeeded|Failed|Unknown>

Important > **kube_pod_status_reason** - <Evicted|NodeAffinity|NodeLost|Shutdown|UnexpectedAdmissionError>

```
Events:
  FirstSeen   LastSeen        Count   From                          SubobjectPath   Type            Reason
  ---------   --------        -----   ----                          -------------   --------        ------
  1m            48s             7       {default-scheduler }                          Warning         FailedScheduling
fit failure on node (kubernetes-node-6ta5): Node didn't have enough resource: CPU, requested: 1000, used: 1420, capacity: 2000
fit failure on node (kubernetes-node-wul5): Node didn't have enough resource: CPU, requested: 1000, used: 1100, capacity: 2000
```

```
Conditions:
  Type              Status    LastHeartbeatTime               LastTransitionTime              Reason              Message
                                                                                                -------
  NetworkUnavailable   False     Thu, 17 Feb 2022 17:09:13 -0500   Thu, 17 Feb 2022 17:09:13 -0500   WeaveIsUp           Weave pod has set this
  MemoryPressure       Unknown   Thu, 17 Feb 2022 17:12:40 -0500   Thu, 17 Feb 2022 17:13:52 -0500   NodeStatusUnknown   Kubelet stopped posting node status.
  DiskPressure         Unknown   Thu, 17 Feb 2022 17:12:40 -0500   Thu, 17 Feb 2022 17:13:52 -0500   NodeStatusUnknown   Kubelet stopped posting node status.
  PIDPressure          Unknown   Thu, 17 Feb 2022 17:12:40 -0500   Thu, 17 Feb 2022 17:13:52 -0500   NodeStatusUnknown   Kubelet stopped posting node status.
  Ready                Unknown   Thu, 17 Feb 2022 17:12:40 -0500   Thu, 17 Feb 2022 17:13:52 -0500   NodeStatusUnknown   Kubelet stopped posting node status.
```

kubectl describe pod XXX

*** Need help? Great links here > Introspection Apps, Troubleshoot Apps, & Debug Running Pods**

Problem - Control Plane

Kubernetes Community Days

Why should you care?

API server - api for all things K8s. Sharing states, and communicate with other K8s components.

API servers went through many iterations of stability. Most problems now related to **inter-network** issues.

etcd as a key-value database - stores the configuration, actual state vs desired state of the system.

Tip - Don't forget **storage**, and **backups** for etcd!

Good guide available by Kubernetes - **HERE**.

K8s Control Plane aka Mission Control

Problem - Control Plane

Kubernetes Community Days

Where to find them? - API Server & etcd

Note > API Server has 3 API endpoints (healthz, livez and readyz).
The healthz endpoint is deprecated (since Kubernetes **v1.16**)
Use the more specific livez and **readyz** endpoints instead.

```
[+]ping ok
[+]log ok
[+]etcd ok
[+]poststarthook/start-kube-apiserver-admission-initializer ok
[+]poststarthook/generic-apiserver-start-informers ok
[+]poststarthook/start-apiextensions-informers ok
[+]poststarthook/start-apiextensions-controllers ok
[+]poststarthook/crd-informer-synced ok
[+]poststarthook/bootstrap-controller ok
[+]poststarthook/rbac/bootstrap-roles ok
[+]poststarthook/scheduling/bootstrap-system-priority-classes ok
[+]poststarthook/start-cluster-authentication-info-controller ok
[+]poststarthook/start-kube-aggregator-informers ok
[+]poststarthook/apiservice-registration-controller ok
[+]poststarthook/apiservice-status-available-controller ok
[+]poststarthook/kube-apiserver-autoregistration ok
[+]autoregister-completion ok
[+]poststarthook/apiservice-openapi-controller ok
healthz check passed
```

kubectl get --raw='/readyz?verbose'

Important > readyz (API Server)

HA with kubeadm

```
etcdctl --write-out=table --endpoints=$ENDPOINTS endpoint status
```

ENDPOINT	ID	VERSION	DB SIZE	IS LEADER	IS LEARNER	RAFT TERM	RAFT INDEX	RAFT APPLIED INDEX	ERRORS
10.240.0.17:2379	4917a7ab173fabe7	3.5.0	45 kB	true	false	4	16726	16726	
10.240.0.18:2379	59796ba9cd1bcd72	3.5.0	45 kB	false	false	4	16726	16726	
10.240.0.19:2379	94df724b66343e6c	3.5.0	45 kB	false	false	4	16726	16726	

```
etcdctl --endpoints=$ENDPOINTS endpoint health

10.240.0.17:2379 is healthy: successfully committed proposal: took = 3.345431ms
10.240.0.19:2379 is healthy: successfully committed proposal: took = 3.767967ms
10.240.0.18:2379 is healthy: successfully committed proposal: took = 4.025451ms
```

Important > health (etcd)

Bonus - Problem - What about my App?

Kubernetes Community Days

Why should you care?

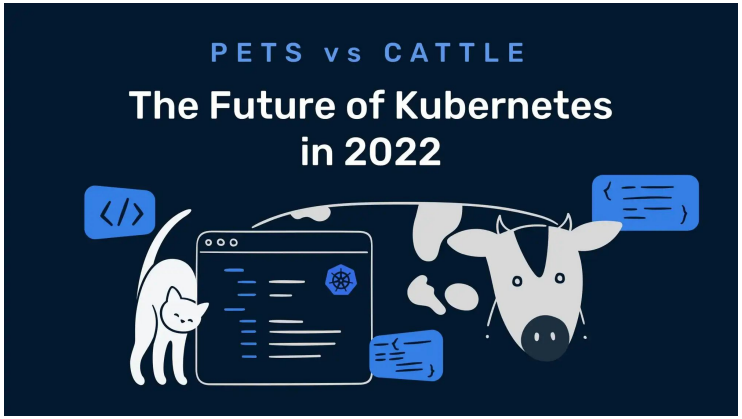
The ideal world in K8s - **stateless** applications.

Monitoring a complex application is a significant endeavor, and each language is unique.

Cats - Servers treated as indispensable or unique systems that can never be down.

Cattle - Servers using automated tools, and are designed for failure, where no one irreplaceable.

[Golden Signals](#) (Google SRE) can be a good start. Latency, Traffic, Errors, and Saturation.



Pets vs Cattle is still relevant in 2022
Source - [2016](#), [Traefik](#)

Bonus - Problem - What about my App?

Kubernetes Community Days

Where to find them?

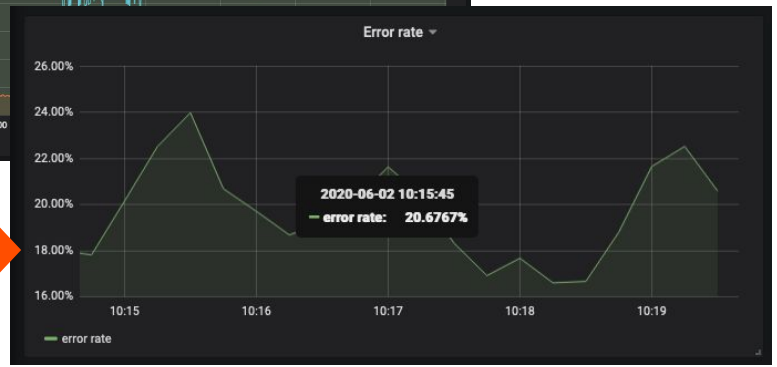
```
package main

import (
        "net/http"

        "github.com/prometheus/client_golang/prometheus/promhttp"
)

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

Prometheus [Go client library](#) to instrument Go applications.

Note - Metrics don't provide deep context.

Important > **95th percentile latency**
The maximum latency, in seconds, for the fastest 95% of requests.

Important > **Error Rate %**
The rate of requests that fail



Recap - what we covered so far …

Kubernetes Community Days

1. Misconfiguration Oops! > Pipeline **Status.**
2. I need MOORRREEEE! > Resources **CPU** and **Memory.**
3. Network Headaches > DNS **Error**, **Health**, and Ingress **Error Rate.**
4. Help! My Pods are stuck! > Status **Phase**, and Status **Reasons.**
5. Control Plane > API Server **Readyz**, and etcd **Health.**
6. What about my Apps? > **P95** and **Error Rate**.

Recap - what we covered so far …

Kubernetes Community Days

Post Mortem is a key process, to understand, mitigate, and prepare.

It's **ALRIGHT** for things to fail. Redirection (Plan B) is needed when things fail.

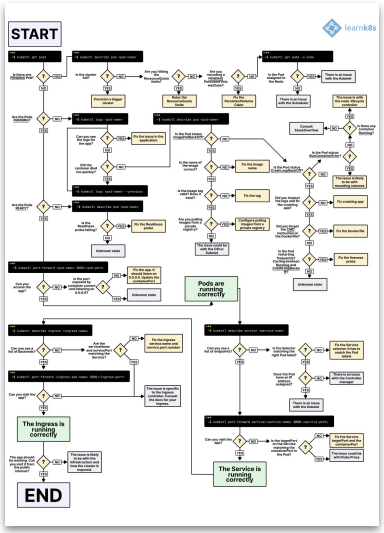
The community is getting better with Kubernetes adoption, and operational flow.

Tempting, but managed K8s isn't perfect. Fundamental understanding is still important!

Trying to improve context relationship between Apps, K8s, and related Pods.

Recap - K8s Troubleshooting Guide

Kubernetes Community Days

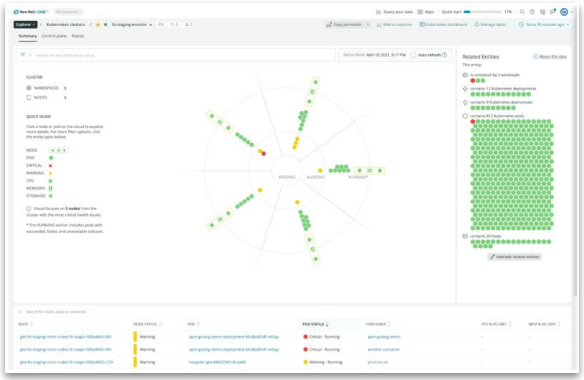


Troubleshooting in Kubernetes can be intimidating if you don't know where to start.

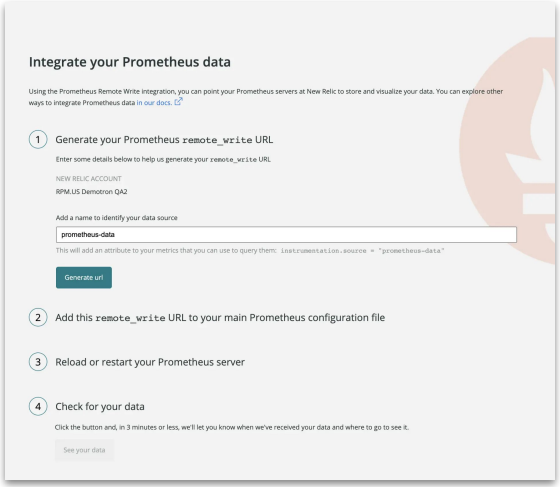
See **HERE**, by **learnk8s**.
See **HERE**, by **Kubernetes.io**.

Kubernetes Observability

Kubernetes Community Days



[Kubernetes Cluster Explorer]



[Managed Prometheus Data]

[Grafana & PromQL Support]

QR Code Here

Kubernetes Community Days

A copy of this deck (including reference links)

Special page for KCD

Reference Links

- cAdvisor
- Metric Server
- API Server
- Node Exporter
- Kube State Metrics

Get started in Kubernetes with New Relic

- Introduction to K8s Integration
- Install & Configure
- Link application to K8s
- Understand K8s Cluster Explorer
- K8s Monitoring Guide