

Kubernetes 導入 Prometheus

Kevin K Chang 張凱傑

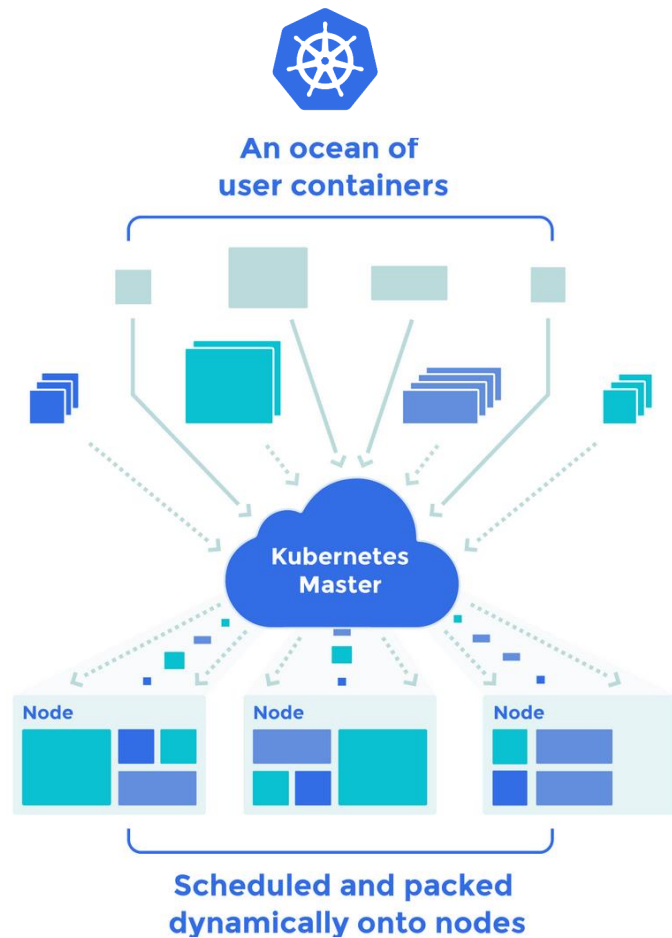
2016 / 9 / 22



Kubernetes

Kubernetes

- Pod / Service / RC
- Resource Management
- Replica control
- Rolling update



Metric retrieve

Dynamic Pod number

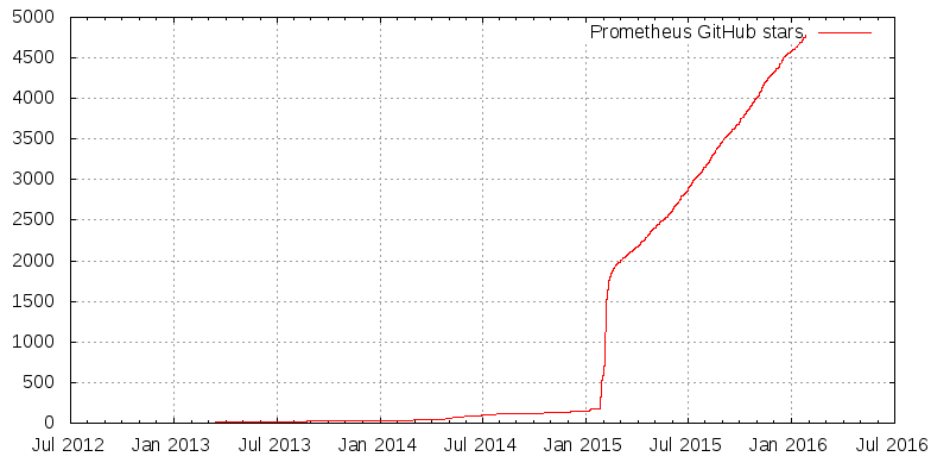


Dynamic Pod IP

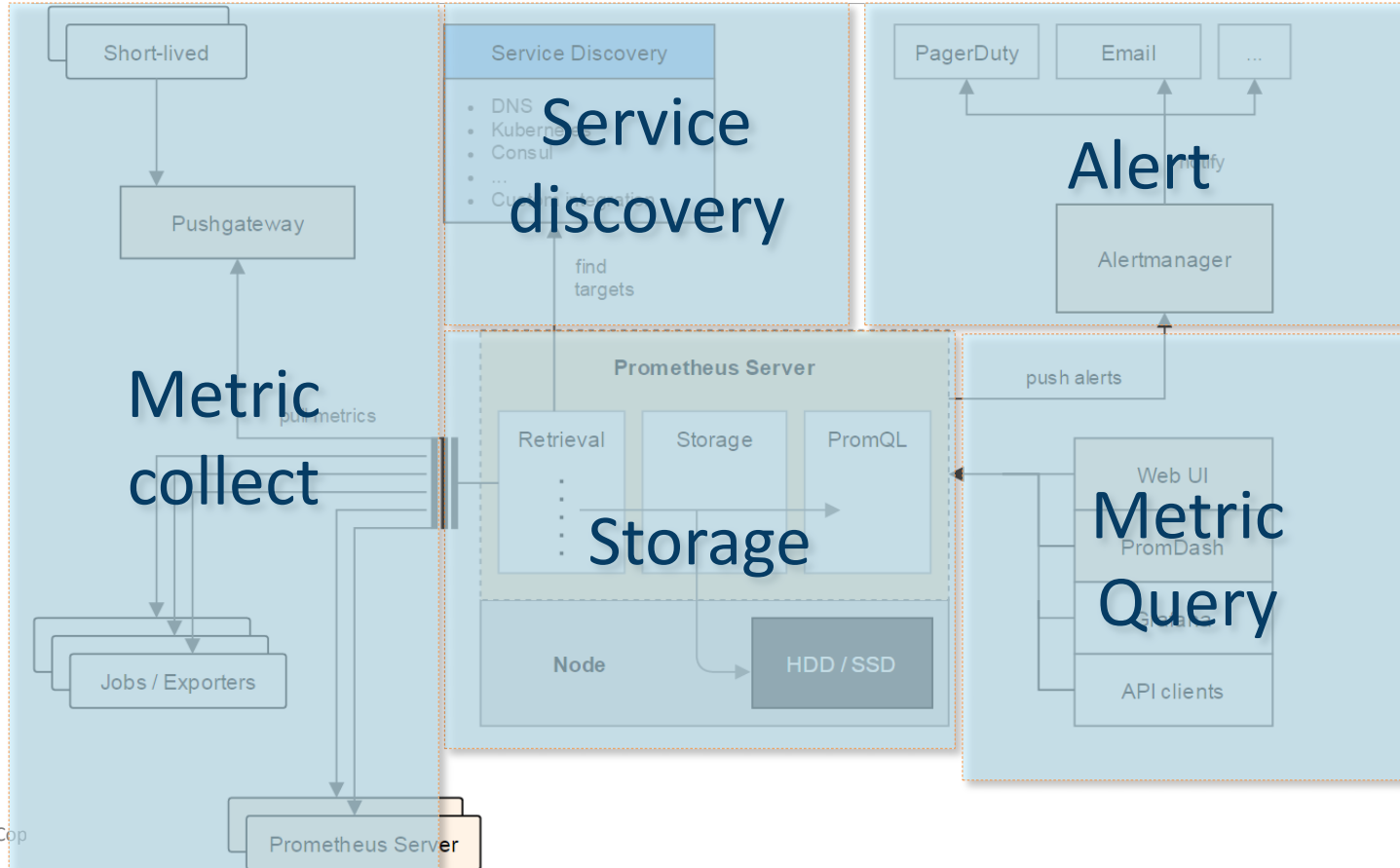
Prometheus

Why Prometheus?

- All in one
- Rich metrics
- Powerful QL
- Easy to integrate



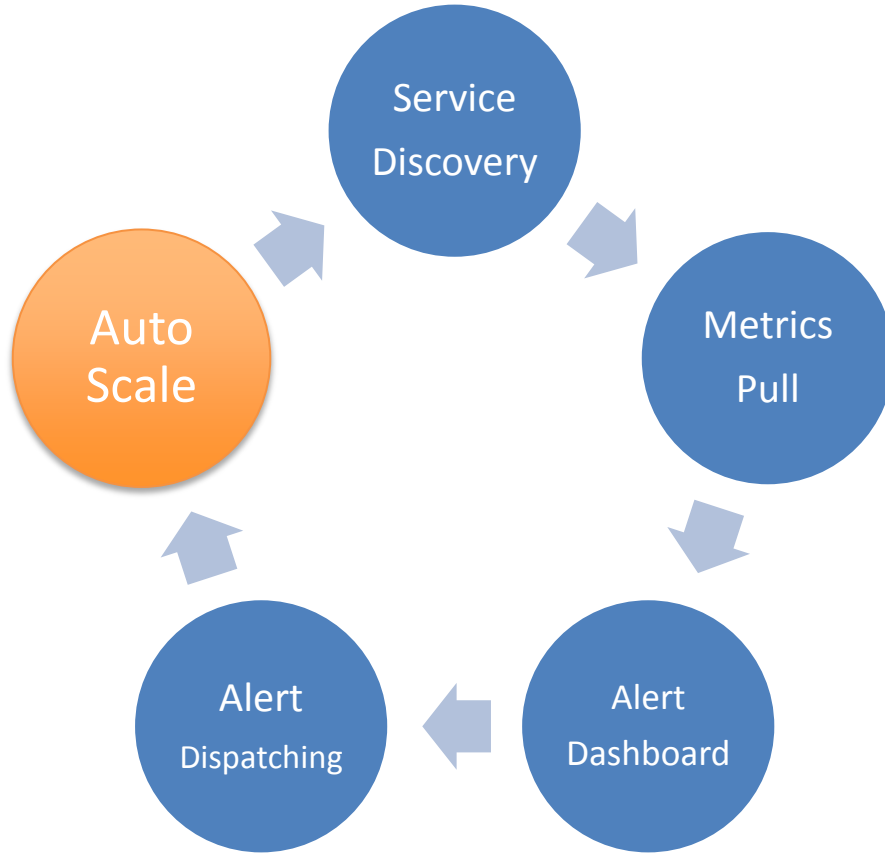
All in One Solution



Prometheus Workflow



Enhanced Workflow



1. Service Discovery

1. Service Discovery

- Support multiple Discovery method
 - DNS / File / Consul / Azure / EC2 / Kubernetes / Marathon / nerve / serverset
- Easy to use and integrate

k8s-cluster

Endpoint	State	Labels	Last Scrape	Error
https://kubernetes.default.svc:443/metrics	UP	none	27.561s ago	
kubernetes-nodes				
Endpoint	State	Labels	Last Scrape	Error
https://10.140.0.10:10250/metrics	UP	<code>beta_kubernetes_io_instance_type="n1-standard-4"</code> <code>cloud_google_com_gke_nodepool="default-pool"</code> <code>failure_domain_beta_kubernetes_io_region="asia-east1"</code> <code>failure_domain_beta_kubernetes_io_zone="asia-east1-a"</code> <code>kubernetes_io_hostname="gke-kube-alpha-default-pool-13f470e6-xrhg"</code> <code>ops="true"</code> <code>vpngw="true"</code>	21.961s ago	
https://10.140.0.12:10250/metrics	UP	<code>beta_kubernetes_io_instance_type="n1-standard-4"</code> <code>cloud_google_com_gke_nodepool="default-pool"</code> <code>failure_domain_beta_kubernetes_io_region="asia-east1"</code> <code>failure_domain_beta_kubernetes_io_zone="asia-east1-a"</code> <code>kubernetes_io_hostname="gke-kube-alpha-default-pool-13f470e6-x1wv"</code>	14.013s ago	
kubernetes-service-endpoints				
Endpoint	State	Labels	Last Scrape	Error
http://10.0.0.15:9122/metrics	UP	<code>alias="10.0.0.15"</code> <code>k8s_app="trafficserver"</code> <code>kubernetes_name="trafficserver"</code> <code>kubernetes_namespace="default"</code>	16.933s ago	
http://10.0.0.7:9090/prom/metrics	UP	<code>alias="10.0.0.7"</code> <code>app="prometheus"</code> <code>kubernetes_name="prometheus"</code> <code>kubernetes_namespace="default"</code> <code>node="ops"</code>	25.19s ago	

Target Endpoints

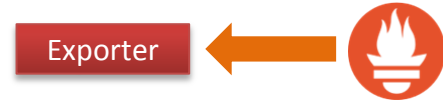
Target Labels

2. Metrics Pull

Exporter



- Metric retrieval logic with HTTP API



Metric name label value

```
node_cpu{cpu="cpu1",mode="system"} 14308.51
node_cpu{cpu="cpu1",mode="user"} 80870.06
```

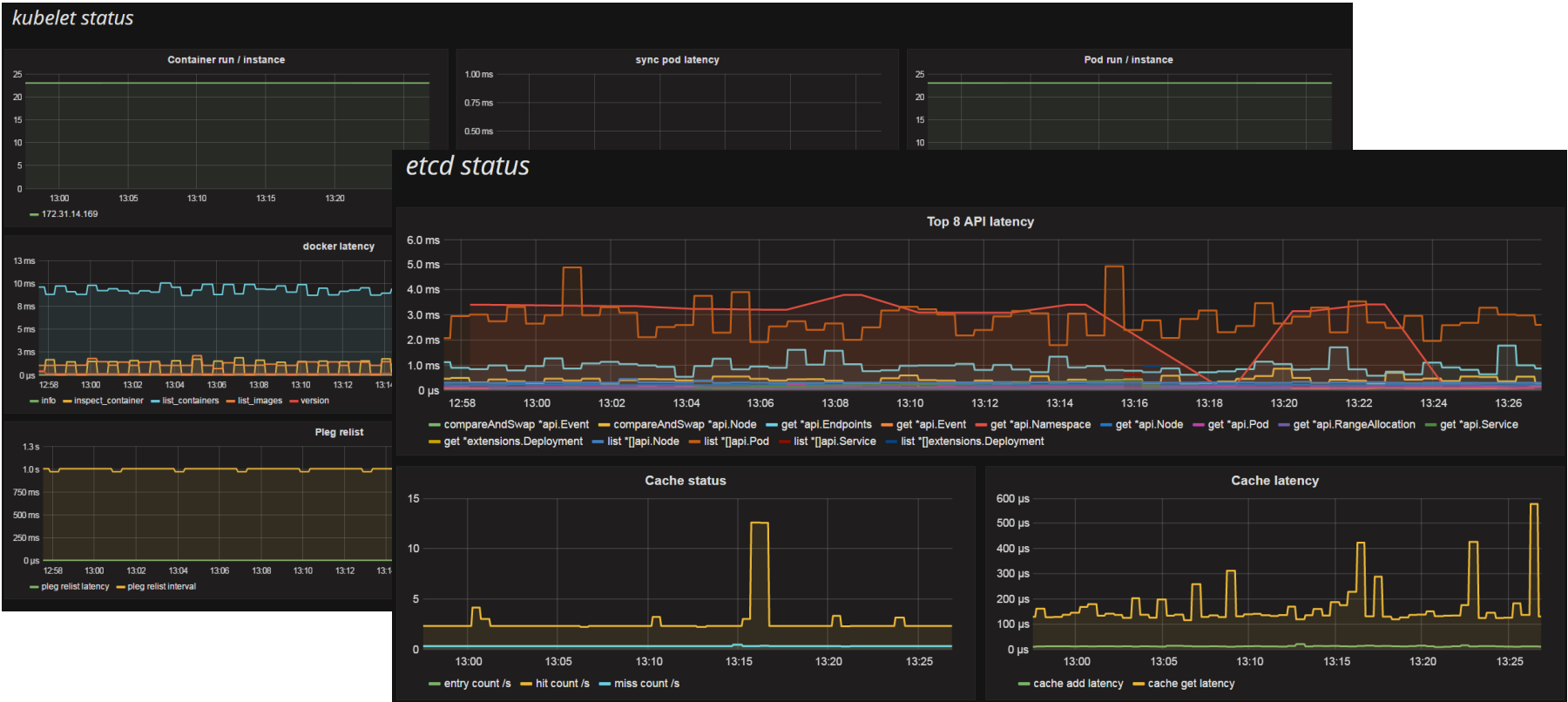
Leverage Existed Exporters

Official third-party exporters

These exporters are maintained as part of the

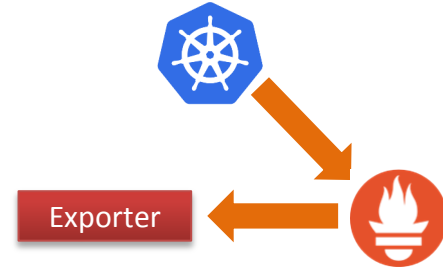
- Node/system metrics exporter
- AWS CloudWatch exporter
- Blackbox exporter
- Collectd exporter
- Consul exporter
- Graphite exporter
- HAProxy exporter
- InfluxDB exporter
- JMX exporter
- Memcached exporter
- Mesos task exporter
- MySQL server exporter
- SNMP exporter
- StatsD exporter
- Aerospike exporter
- Apache exporter
- BIG-IP exporter
- BIND exporter
- Ceph exporter
- CouchDB exporter
- Django exporter
- Google's mtail log data extractor
- Grok exporter
- Heka dashboard exporter
- Heka exporter
- IoT Edison exporter
- IPMI exporter
- Jenkins exporter
- knxd exporter
- Meteor JS web framework exporter
- Minecraft exporter module
- Mirth Connect exporter
- MongoDB exporter
- Munin exporter
- NATS exporter
- New Relic exporter
- Nginx metric library
- NSQ exporter
- OpenWeatherMap exporter
- Passenger exporter
- PgBouncer exporter
- PostgreSQL exporter
- PowerDNS exporter
- RabbitMQ exporter
- RabbitMQ Management Plugin exporter
- Rancher exporter
- Redis exporter
- RethinkDB exporter
- rTorrent exporter
- scollector exporter
- SMTP/Maildir MDA blackbox prober
- Speedtest.net exporter
- SQL query result set metrics exporter
- Ubiquiti UniFi exporter
- Varnish exporter
- WebDriver exporter
- Xen exporter
- Zookeeper exporter

Kubernetes API server/ kubelet Supported



Expose to Prometheus

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: nginx-svc
5  labels:
6    app: nginx
7  annotations:
8    prometheus.io/scrape: "true"
9    prometheus.io/port: "9113"
10   prometheus.io/path: "/metrics"
11  spec:
12    type: NodePort
13  selector:
14    name: nginx-test
15  ports:
16  - name: http
17    port: 80
```



3. Dashboard Alert

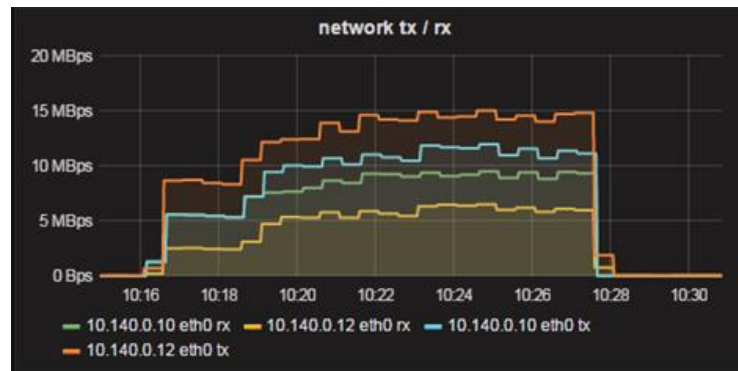
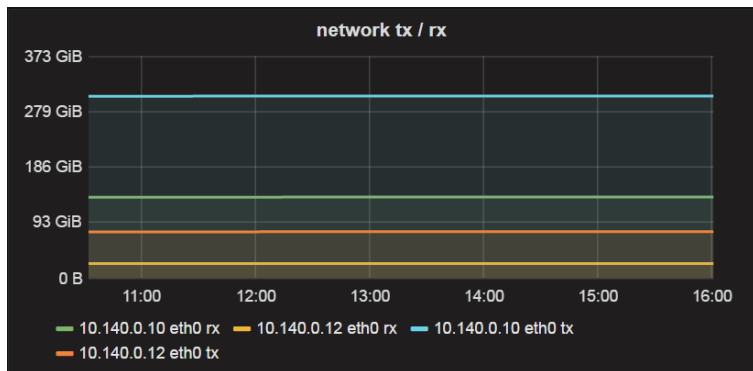
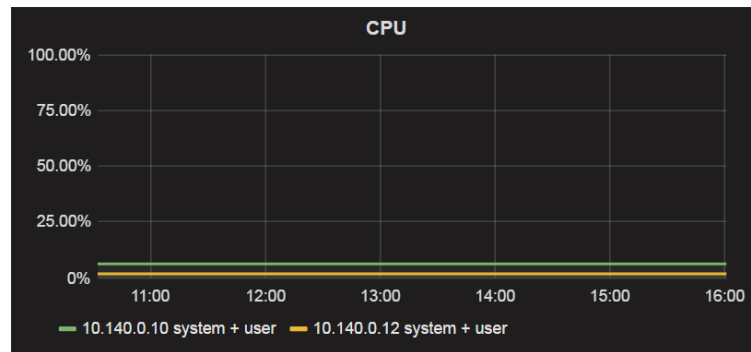
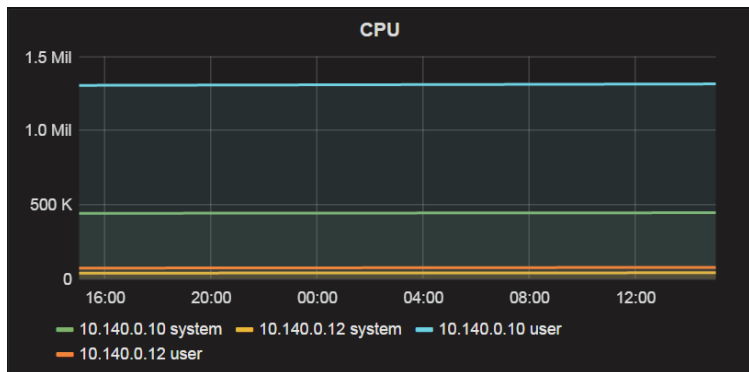
Dashboard - Grafana

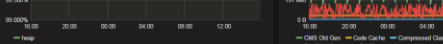
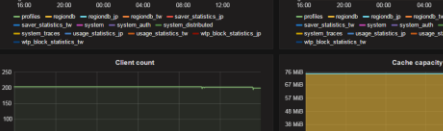
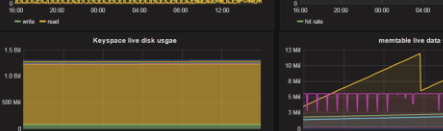
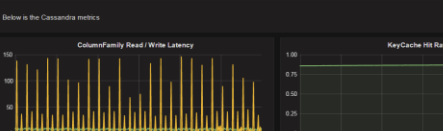
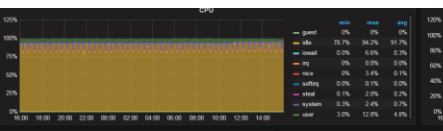
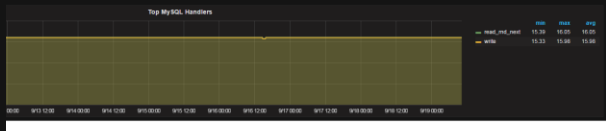
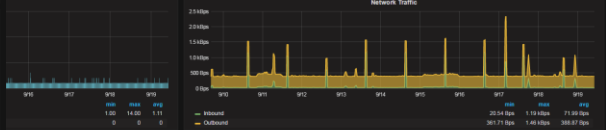
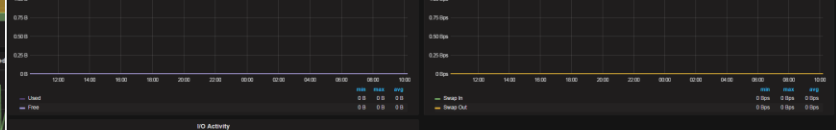
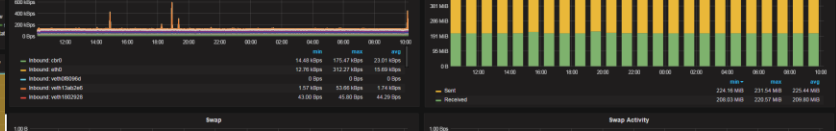
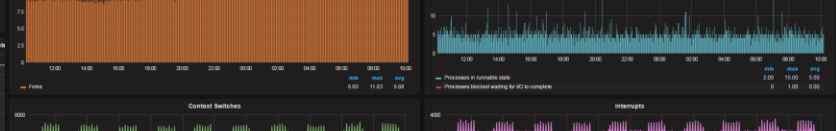
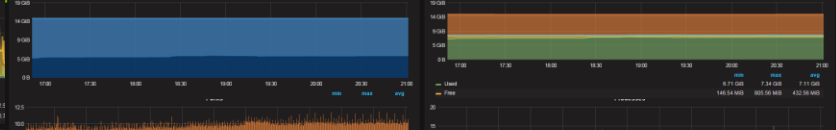
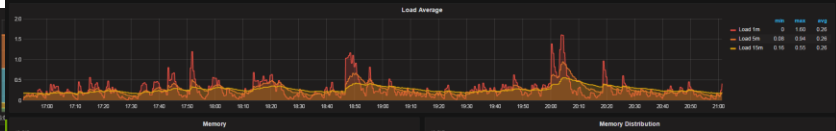
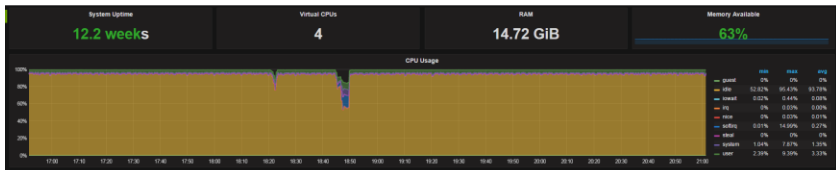
- Generate beautiful / human-readable metric dashboard
- Insight your metrics
- Recommended from Prometheus

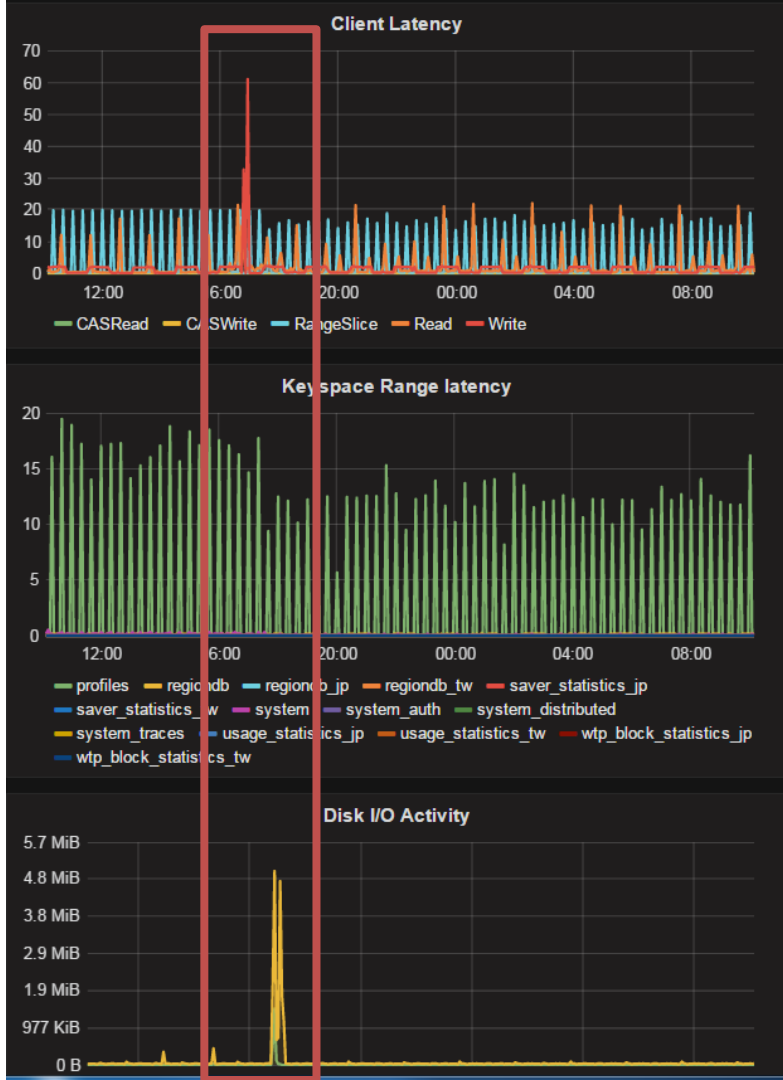
NOTE: We recommend [Grafana](#) for visualization of Prometheus metrics nowadays, as it has native Prometheus support and is widely adopted and powerful. There will be less focus on PromDash development in the future.

Before

After







Alert Format

ALERT <alert name>

IF <expression>

[FOR <duration>]

[LABELS <label set>]

[ANNOTATIONS <label set>]

Create Alert rule for capacity

```
ALERT NGINX_LOAD_HIGH
```

```
IF sum(rate/nginx_connections_processed_total{stage="handled"}[1m])) /  
((count/nginx_connections_processed_total{stage="handled"})) * 100 > 1
```

```
FOR 30s
```

```
LABELS {pager="webhook", service="nginx", severity="critical", svc="nginx"}
```

```
ANNOTATIONS {
```

```
description="{
```

```
  \"rc\": \"nginx-test-rc\",
```

```
  \"ns\": \"default\",
```

```
  \"min\": 2,
```

```
  \"max\": 10,
```

```
  \"scale\": \"out\"},
```

```
summary="nginx load high"}
```


4. Alert Dispatching

Alert Dispatching

- Support routing by label
- Support Email / HipChat / Slack / PagerDuty / Pushover / OpenGenie / Webhook



Setup Alert Routing

routes:

...

- match:

 pager: webhook

 receiver: webhook-k8s

receivers:

- name: "webhook-k8s"

 webhook_configs:

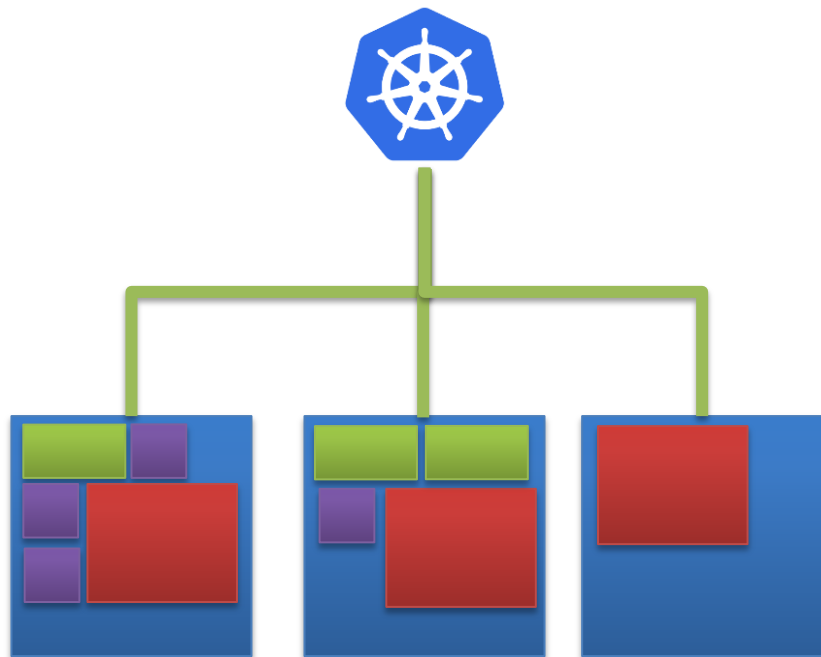
 - url: 'http://k8s-webhook-svc/webhooks/prom'

 send_resolved: true

5. Auto Scale

Auto Scale Kubernetes

- Alert -> Webhook -> K8S
- Scale pod / minion



Send to Custom Webhook

Let's see demo video

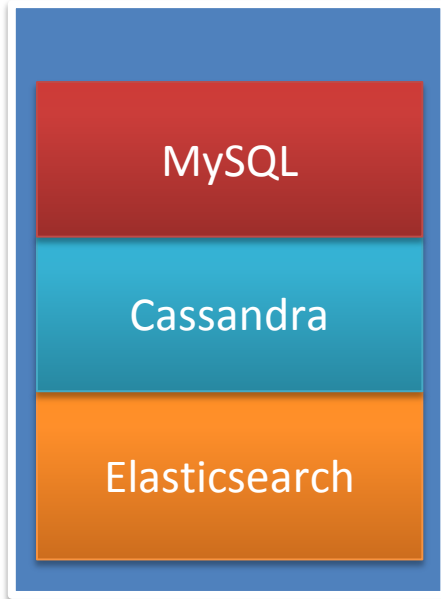
Conclusions

- Easy to integrate
- Great metrics system

Tips sharing

Separate ops / service minions

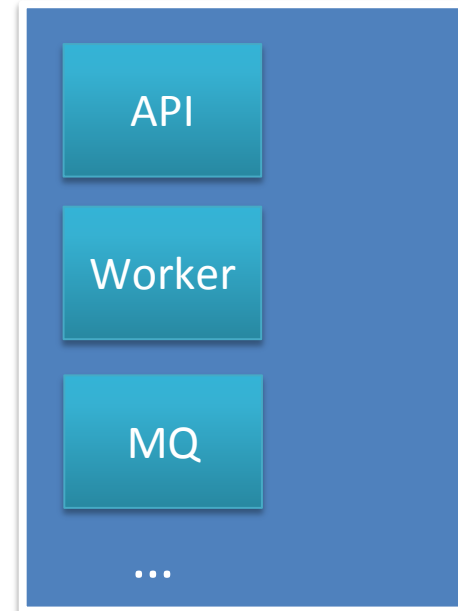
External node



Ops minion / node



service minion



Group your metrics 1/2

```
node_cpu{alias="172.31.23.64",instance="172.31.23.64:9100",job="172.31.23.64:9100",system="node" mode="system"}
cassandra_cache_hitrate{cache="RowCache",instance="172.31.23.64:7070",job="cassandra"}
cassandra_cache_hitrate{alias="172.31.23.64",cache="RowCache",instance="172.31.23.64:7070",job="cassandra"}
```

```
relabel_configs:
- source_labels: [__address__]
  action: replace
  regex: (.+)(?::\d+)
```

```
target_label: alias
replacement: $1
```

Add useful label for filtering

Ex: node="cassandra"

BUT DON'T OVER USE

Group your metrics 2/2

apiVersion: v1

kind: Service

metadata:

annotations:

prometheus.io/scrape: "true"

prometheus.io/port: "9100"

prometheus.io/path: "/metrics"

name: prometheus-node-exporter

labels:

node: cassandra

Thank you

References

- Prometheus Kubernetes configuration
 - <https://github.com/prometheus/prometheus/blob/master/documentation/examples/prometheus-kubernetes.yml>
- Webhook for Kubernetes scale example
 - <https://github.com/kaija/k8s-scale-webhook>