# Grafana optimization for Prometheus

### About me

- Mitsuhiro Tanda
- Infrastructure Engineer @ GREE
- Use Prometheus on AWS (1.5 year)
- Grafana committer
- @mtanda

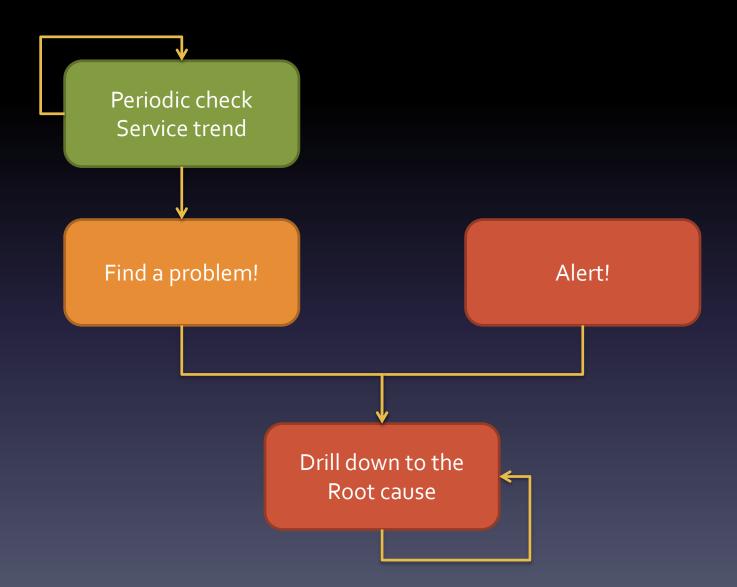
#### Our environment

- deploy multiple Prometheus for each service
- each service launch 100 or more instances
- rarely use RDS, run MySQL on EC2
- various service, role, and many instances

## Dashboard policy

- Adapt dynamic environment with Auto scaling
- Avoid service specific parameter hard coding
- Reuse same dashboard for several service
- Prepare dashboard for drilldown analysis

# Operation flow



## Key Grafana feature

- Templating
  - Query parameter
  - Datasource
- Panel Repeat
- Scripted dashboard
- Table panel (with Annotations)

# Templated queries

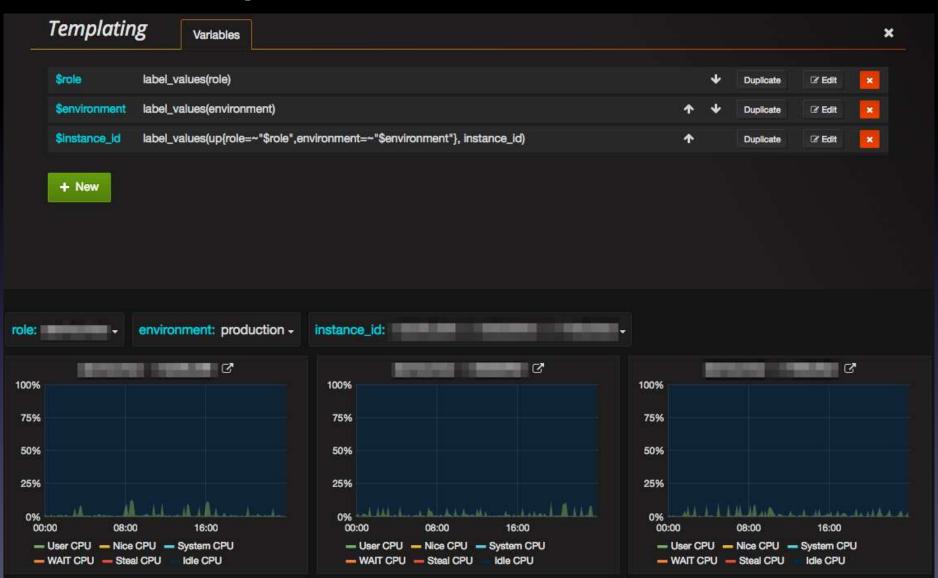
Name	Description
label_values(label)	Returns a list of label values for the label in every metric.
label_values(metric, label)	Returns a list of label values for the label in the specified metric.
metrics(metric)	Returns a list of metrics matching the specified metric regex.
query_result(query)	Returns a list of Prometheus query result for the query.

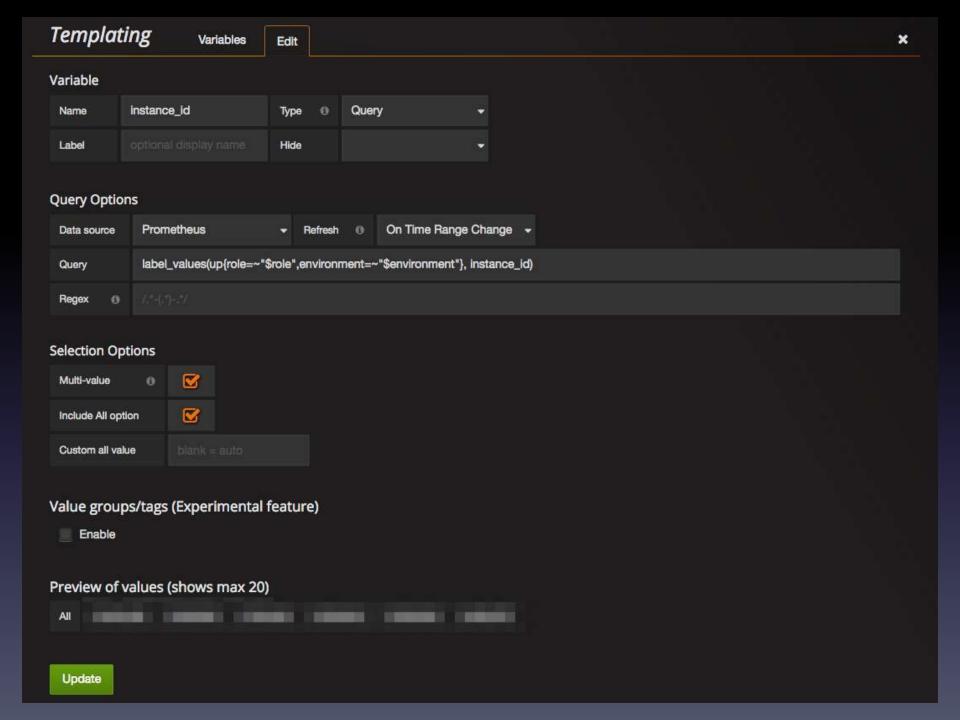
http://docs.grafana.org/datasources/prometheus/

### Service trend

- Prepare dashboard for key metrics
  - CPU Utilization
  - Response time
  - Etc.
- Filter by role, and check deeply

## Dynamic dashboard

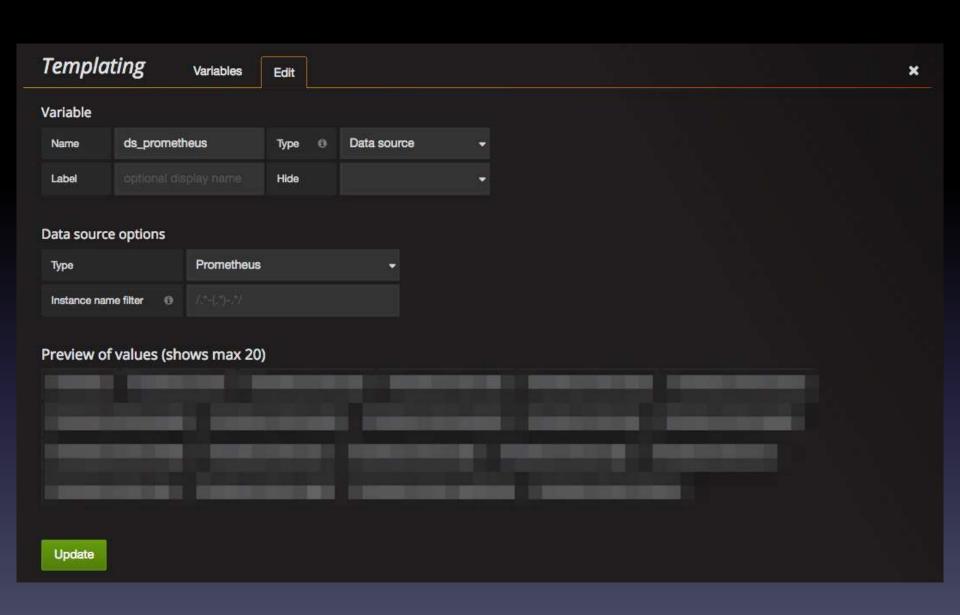




- Refresh option
  - "On Time Range Change"
  - Query each time when time range changed
- label\_values(metrics, label\_key)
  - Get label values from metrics
  - Only match the metrics in current time range
  - (match current active instances)

## Datasource templating





- Switch datasource quickly
- Can check several service on same dashboard

#### Alert

- We use PagerDuty to call on-call engineer
- Alert message also be posted to chat
- Message contains shortcut link to alert dashboard

## Prometheus alert view

```
MysqlSlaveDelay (0 active)

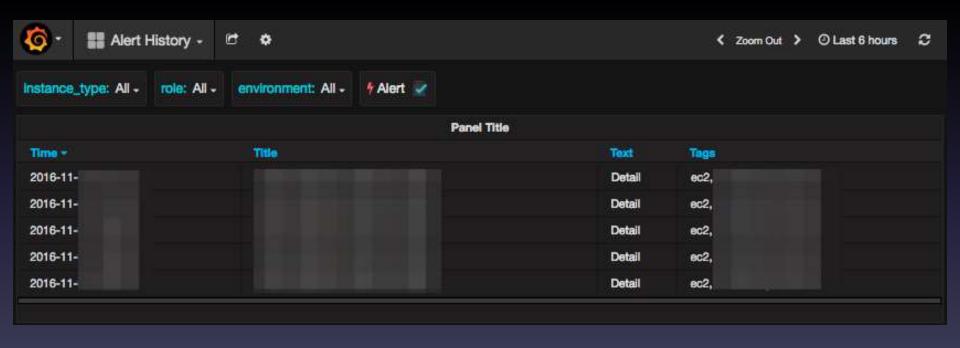
ALERT MysqlSlaveDelay
   IF mysql:my_seconds_behind_master{role=~"(mysql|rds)-slave.*"} >= 10
   FOR 1m
   ANNOTATIONS {description="Mysql Slave Delay Alert (10 sec) (current value: {{$value}})"}
```

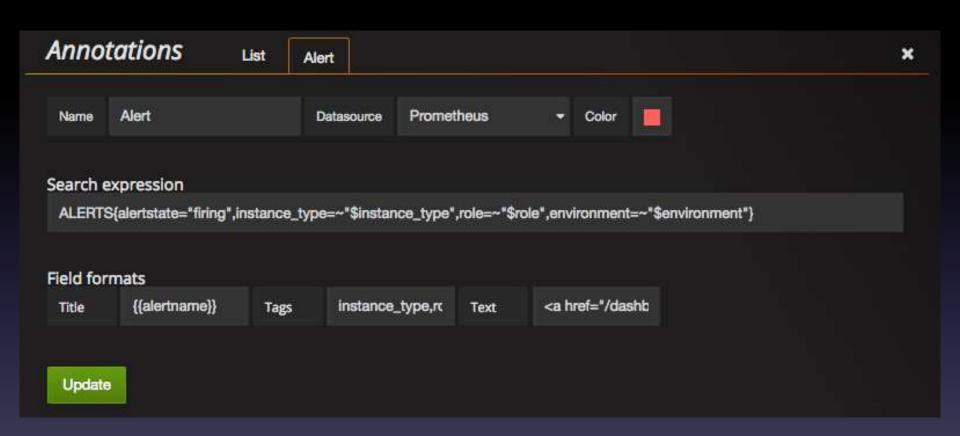
#### Grafana alert view

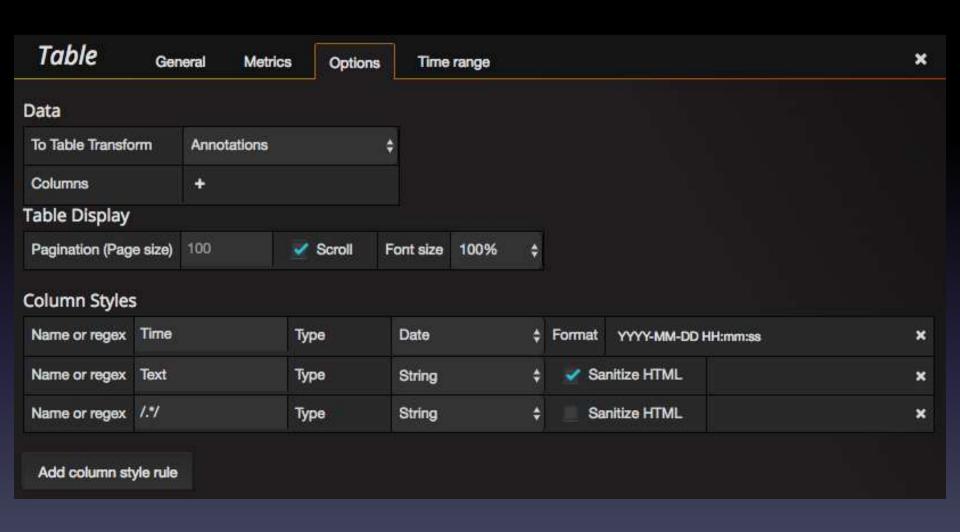


- Use Scripted dashboard
- Parse Prometheus alert view HTML
- And generate dashboard
- https://gist.github.com/mtanda/2abaoe96d2a8aace7b6b9a9o3bcd6b31

# Alert history







- Query "ALERTS" metrics of Prometheus
- Set alert annotation data to Table panel

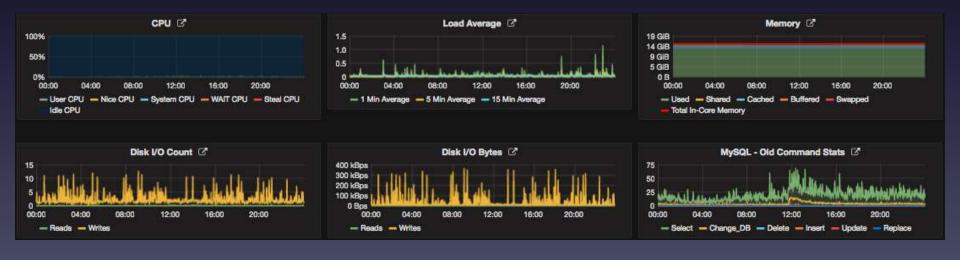
#### Drilldown

- Show graphs for corresponding instance roles
- Host level metrics and systems metrics
- Need to create dashboard dynamically
- Use Scripted dashboard



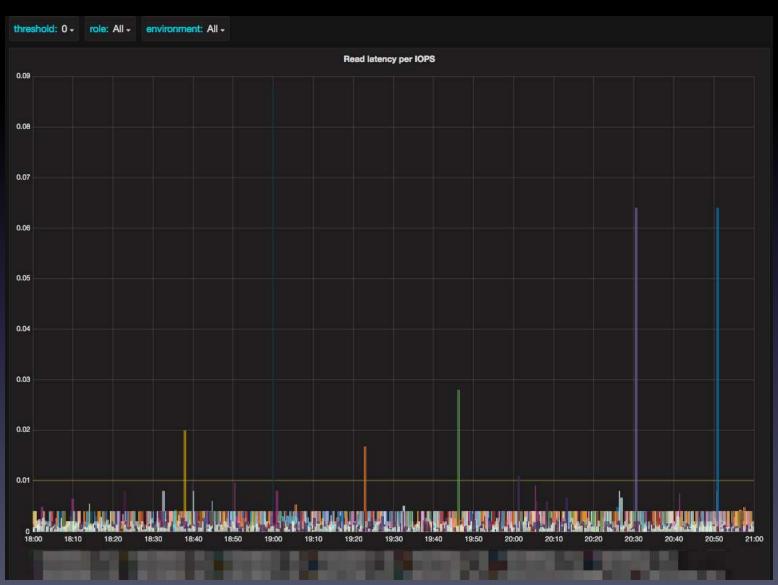


Generate dashboard!



```
"title": "MySQL - Old Command Stats",
"stack": false,
"fill": 0,
"yaxes": [
    "show": true,
    "min": 0,
    "max": null,
    "format": "short"
"seriesOverrides": [
    "alias": "Select",
    "fill": 2
"targets": [
  { "expr": "mysql:my com_select#L#", "legendFormat": "Select" },
 { "expr": "mysql:my_com_change_db#L#", "legendFormat": "Change_DB" },
 { "expr": "mysql:my_com_delete#L#", "legendFormat": "Delete" },
  { "expr": "mysql:my_com_insert#L#", "legendFormat": "Insert" },
 { "expr": "mysql:my_com_update#L#", "legendFormat": "Update" }.
  { "expr": "mysql:my_com_replace#L#", "legendFormat": "Replace" }
```

## EBS latency dashboard



- Filter by role and threshold
- Quickly find problematic instances

## Wrap up

- Grafana is very powerful visualization tool
- It is little tricky, but very flexible
- Make better Grafana by contributing!