ASSIGNMENT

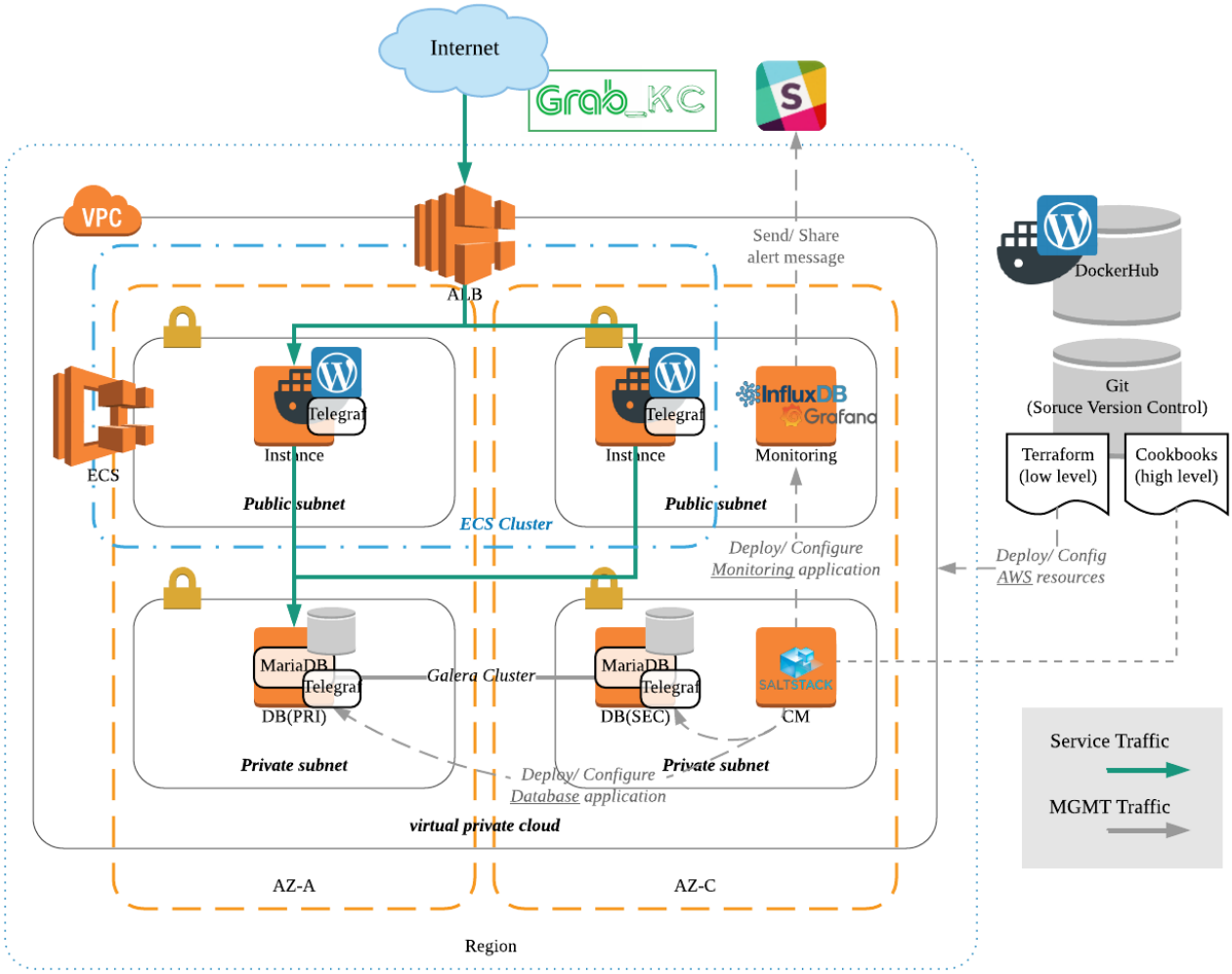
ASSIGNMENT

INTRODUCTION

I developed Terraform to deploy AWS low-level building block infrastructure such as VPC, ALB, and ECS. I also used Saltstack to develop application cookbooks as configuration management and role-based deployment to applications such as MariaDB, Galera clusters, and TIK stacks (Telegraf, InfluxDB, Kapacitor).

Designed VPC which has public/private subnets, and we can protect the application from maleficent traffic. The ECS cluster is on multi-AZ for HA and configured with ASG for application availability and elasticity. In addition, I developed a CM structure with scalability and reusable in order to serve a lot of GRAB services in the same cookbooks. In the monitoring system, I defined docker, system, and MariaDB monitoring metrics and make Grafana templates. lastly, I integrated the Kapacitor service to slack in order to get the alert messages in the Ops slack channel immediately.

ARCHITECTURE



IMPLEMENTATION

1. Deploy low-level infrastructure such as networking (VPC, Subnet, Routing, and etc), ECS, and EC2 using Terraform.
2. Integrate SaltStack with private cookbooks on GitLab for role-based deployment and manage application(Database, Monitoring) configuration.
3. Configure Telegraf container for gathering system and application monitoring metrics to InfluxDB.
4. Share visible monitoring metrics via Grafana and get alert message in Ops Slack channel through Kapacitor.

RESULT

1. Implement the architecture using Terraform

   Source file:

   URL: https://github.com/cloudacode/runbook-interview

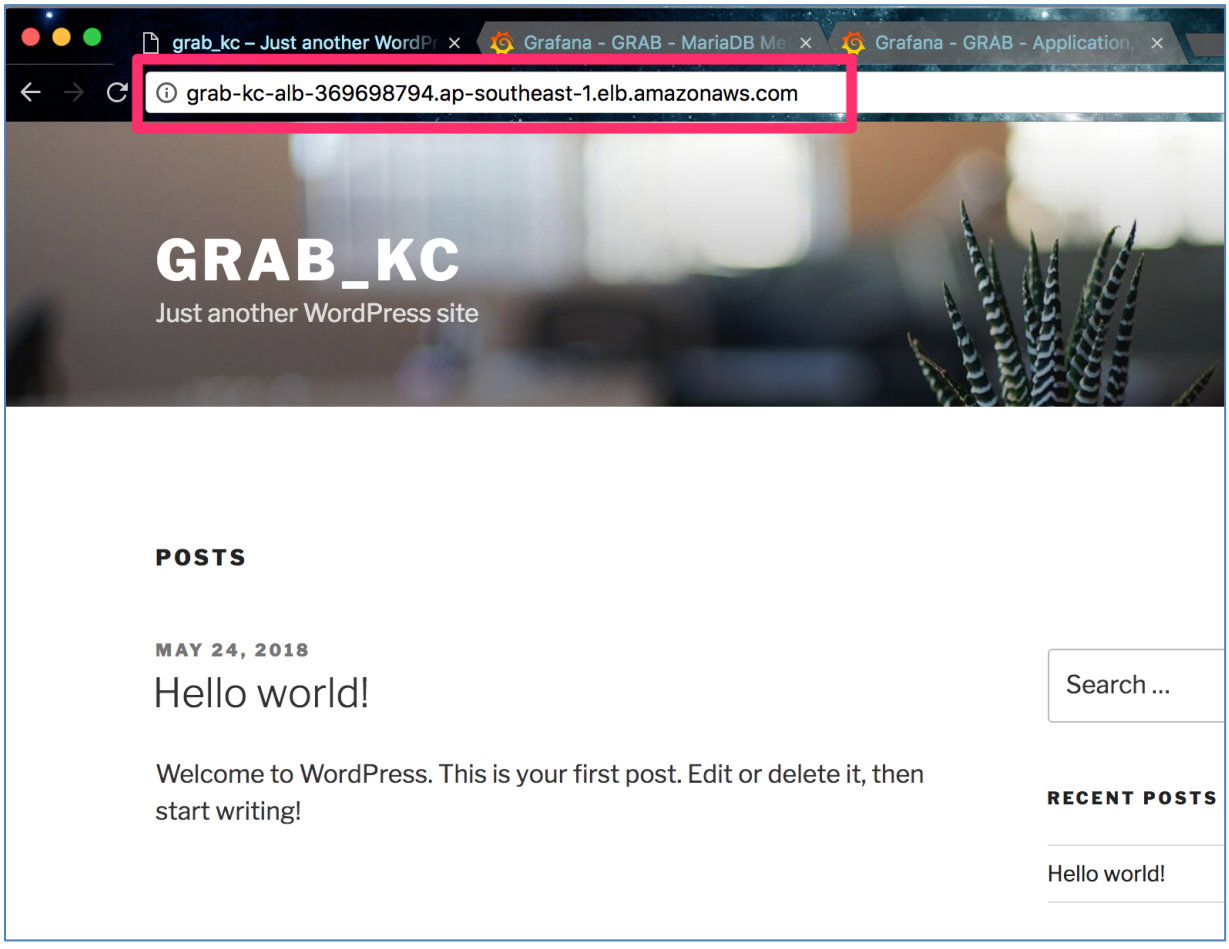
   Console output:

```
  load_balancer.1688621339.elb_name:          "" => ""
  load_balancer.1688621339.target_group_arn: "" => "arn:aws:elasticloadbalancing:ap-southeast-1:255171805824:targetgroup/kc-g
  name:                                       "" => "kc-grab18-service"
  task_definition:                            "" => "arn:aws:ecs:ap-southeast-1:255171805824:task-definition/kc-grab18-td:2"
aws_ecs_service.grab-service: Creation complete after 0s (ID: arn:aws:ecs:ap-southeast-1:255171805824:service/kc-grab18-servi

Apply complete! Resources: 38 added, 0 changed, 0 destroyed.

Outputs:

alb_url = http://kc-grab18-alb-1900891948.ap-southeast-1.elb.amazonaws.com
db_master_private_ip = 10.200.3.41
db_slave_private_ip = 10.200.2.82
ecs_name = kc-grab18-cluster
monitoring_server_ip = 52.77.248.208
monitoring_url = http://52.77.248.208:3000
salt_master_log_in = ssh -i ~/environment/grab-kc-key.pem ubuntu@52.221.187.90
salt_master_server_ip = 52.221.187.90
update_slatstack_pillar1 = sed -i 's/node01_ip/10.200.3.41/g' /srv/pillar/service/grab_kc/mariadb.sls
update_slatstack_pillar2 = sed -i 's/node02_ip/10.200.2.82/g' /srv/pillar/service/grab_kc/mariadb.sls
ec2-user:~/environment/kc-grab-project/terraform (master) $
```



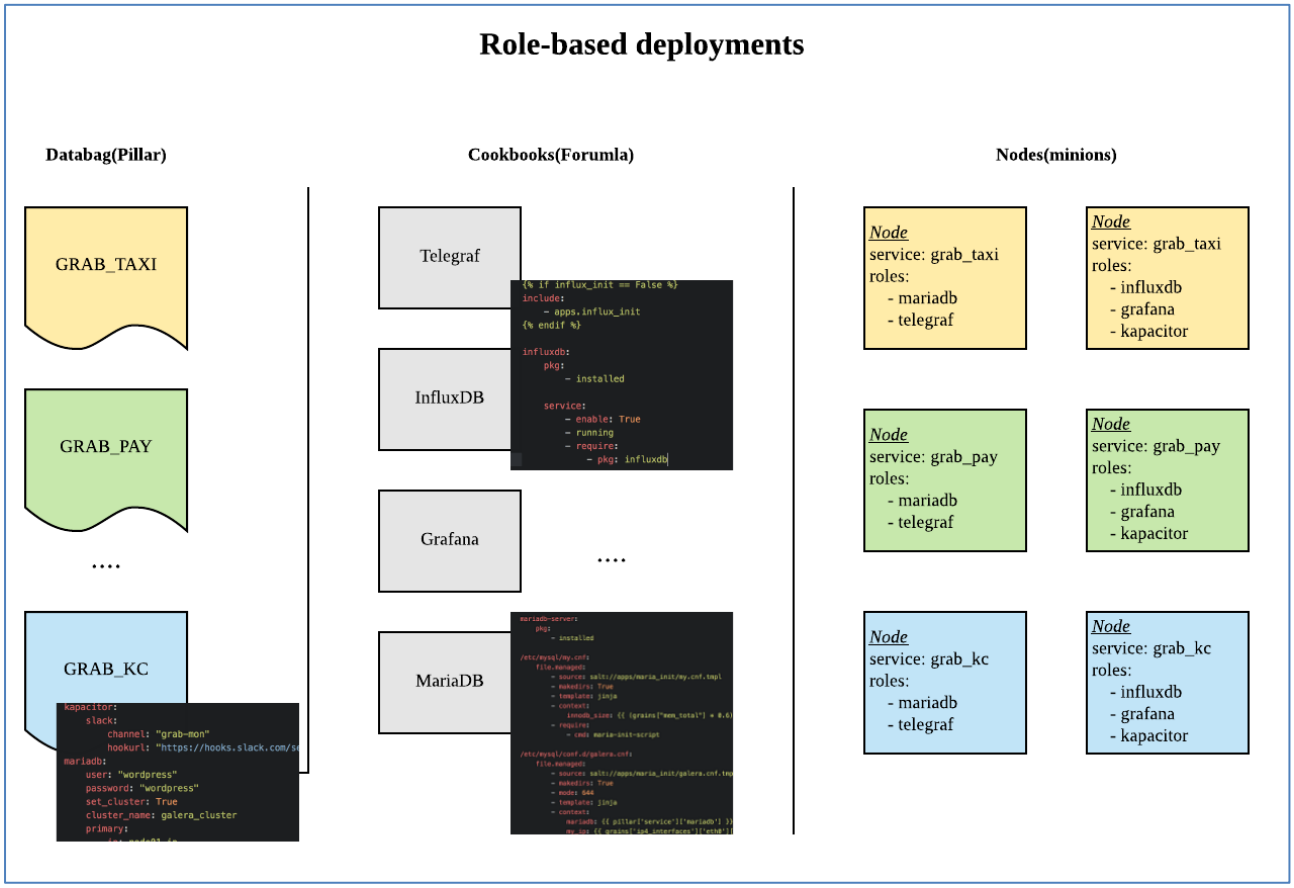
2. Integrate SaltStack to deploy and manage applications.

   Source file:

   GitHub URL: https://github.com/cloudacode/runbook-interview

Service Architecture:

- Databag(Pillar): A global variable of the service. The value defines service metadata that is assigned to one or more nodes.
- Cookbooks(Fourmla): A cookbook(formula) defines a scenario of process task such as installing a package, configuring, and starting a service.



Console output:

3. Deploy Wordpress container image with Telegraf monitoring solutions.

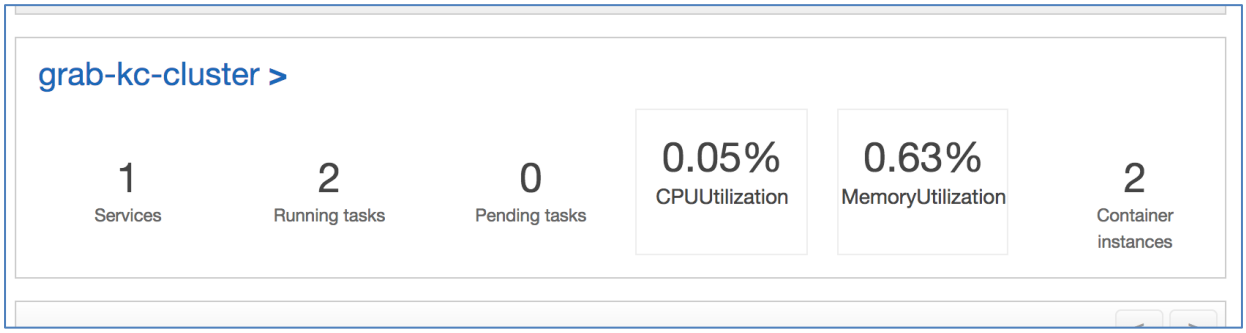
Source file:

URL: https://github.com/cloudacode/runbook-interview/blob/main/terraform/wp-task-definition.json

```
"cpu": 128,
"essential": true,
"image": "${wordpress_image_url}",
"memory": 256,
"memoryReservation": null,
"name": "${wordpress_container_name}",
"portMappings": [
  {
    "hostPort": 80,
    "protocol": "tcp",
    "containerPort": 80
  }
],
"command": null,
"linuxParameters": null,
"environment": [
  {
    "name": "WORDPRESS_DB_HOST",
    "value": "mysql"
  },
  {
    "name": "WORDPRESS_DB_PASSWORD",
    "value": "${wordpress_password}"
  },
```

```
"dnsServers": null,
"mountPoints": [
  {
    "readOnly": true,
    "containerPath": "/var/run/docker.sock",
    "sourceVolume": "dockersocket"
  }
],
"workingDirectory": null,
"dockerSecurityOptions": null,
"memory": 128,
"memoryReservation": null,
"volumesFrom": [],
"image": "${telegraf_image_url}",
"disableNetworking": null,
"essential": true,
"links": [
  "${wordpress_container_name}:wordpress"
],
"hostname": "telegraf",
"extraHosts": [
  {
    "ipAddress": "${influxdb_server_ip}",
    "hostname": "influxdb"
  }
```

Console output



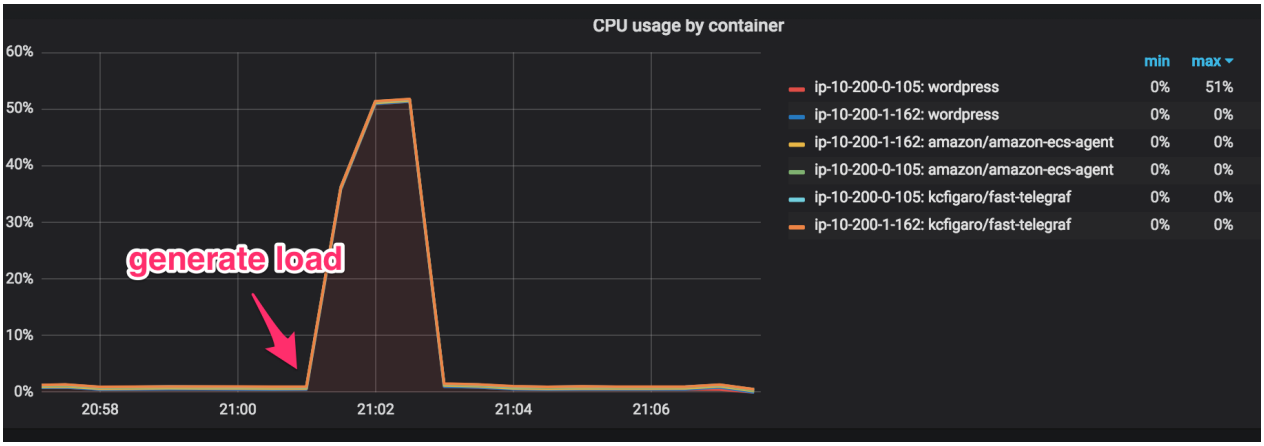
4. Define Grafana web dashboard and get alert message in Ops Slack channel

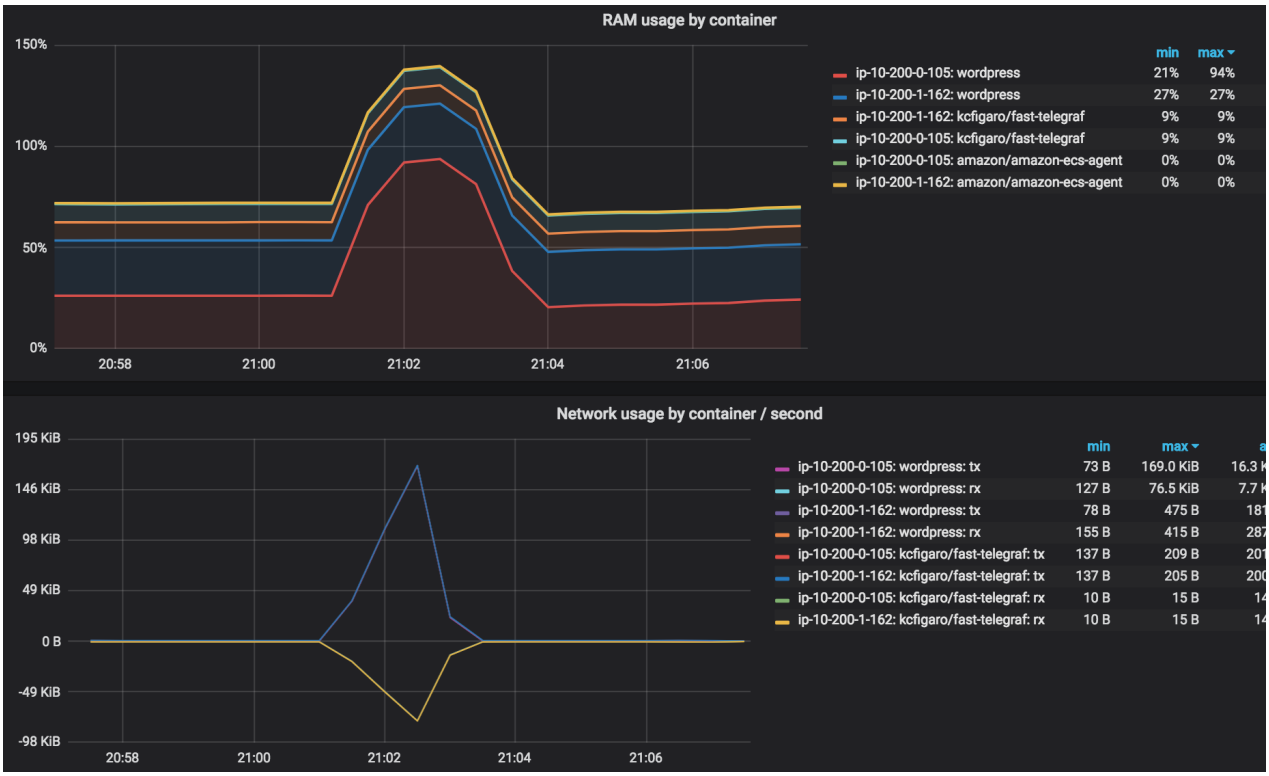
Sample Templetes:

URL: https://github.com/cloudacode/runbook-interview/tree/main/terraform/grafana_template

Grafana Dashboard:

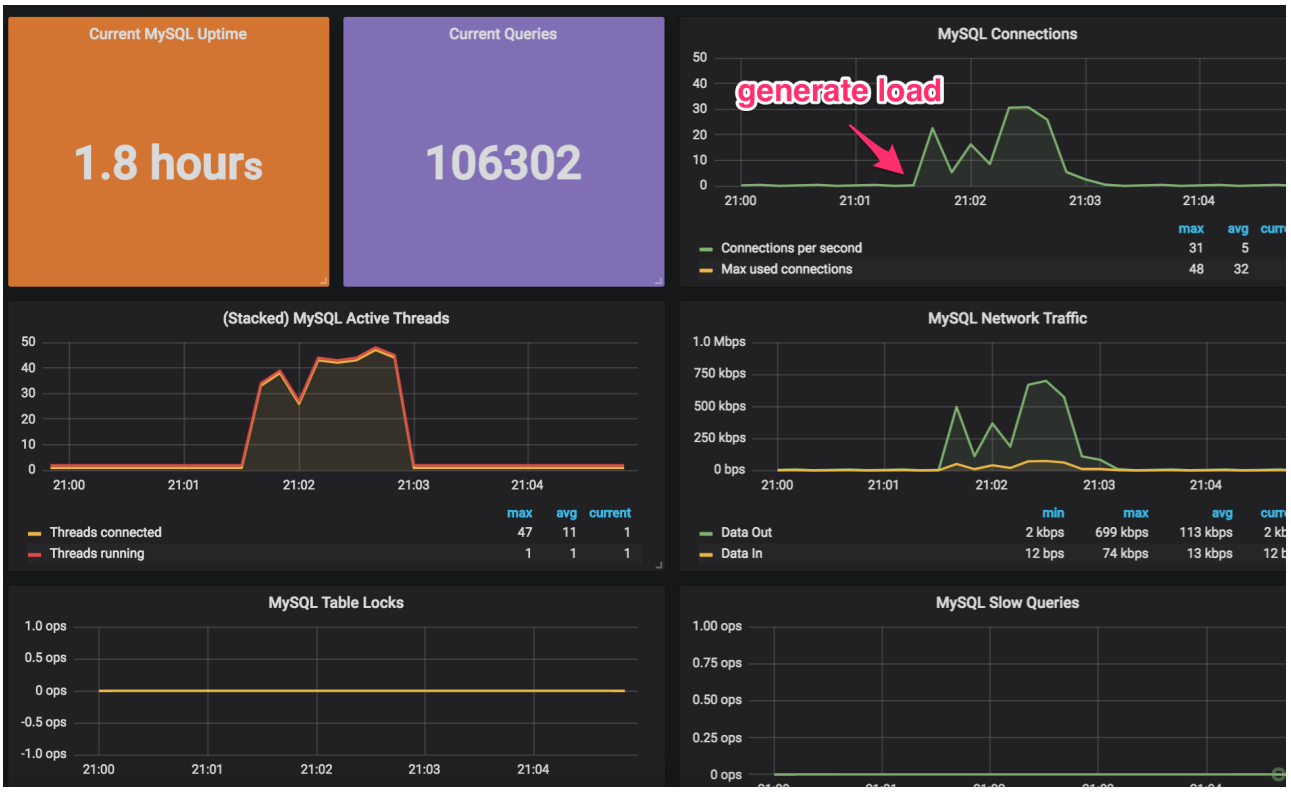
- Application(Container) usage metrics





- System usage metrics

- Database metrics



- Detect and forward alert to Slack Ops channel

```
}
root@ip-10-200-1-83:/data# kapacitor list tasks
ID                          Type      Status    Executing Databases and Retention Policies
docker_container_cpu_usage stream    enabled   true      ["telegraf"."autogen"]
```

```
var warn_threshold = 10
var crit_threshold = 50

dbrp "telegraf"."autogen"
    stream
        |from()
            .measurement('docker_container_cpu')
            .groupBy('engine_host')
            .where(lambda: "container_image" == 'wordpress')
        |window()
            .period(10s)
            .every(10s)
        |alert()
            .id('{{ index .Tags "engine_host"}}/wordpress - usage_percent')
            .message('{{ .ID }} is {{ .Level }} (VAL:{{ index .Fields "usage_percent" | printf "
            .warn(lambda: "usage_percent" > 10)
            .crit(lambda: "usage_percent" > 50)
        .slack()
```

grab-mon APP 1:40 PM
- ip-10-200-0-170/wordpress - usage_percent is CRITICAL (VAL:96.016)
- ip-10-200-0-170/wordpress - usage_percent is CRITICAL (VAL:51.819)
- ip-10-200-0-170/wordpress - usage_percent is CRITICAL (VAL:51.230)
- ip-10-200-0-170/wordpress - usage_percent is WARNING (VAL:44.233)

kcfigaro 1:43 PM
@OPs Please check the ip-10-200-0-170/wordpress status ASAP.

grab-mon APP 1:43 PM
- ip-10-200-0-170/wordpress - usage_percent is WARNING (VAL:54.825)
- ip-10-200-0-170/wordpress - usage_percent is CRITICAL (VAL:104.263)
- ip-10-200-0-170/wordpress - usage_percent is WARNING (VAL:51.246)

OPs 1:44 PM
@kcfigaro CPU usage is high because of unexpected network traffic!!

grab-mon APP 1:44 PM
- ip-10-200-0-170/wordpress - usage_percent is WARNING (VAL:51.355)
- ip-10-200-0-170/wordpress - usage_percent is WARNING (VAL:60.678)
- ip-10-200-0-170/wordpress - usage_percent is OK (VAL:0.003)

kcfigaro 1:57 PM
uploaded this image: 1527137815983.png