promethues

```
Date: 20191204
Author: LiuZhangshu
Version: v1.0.0
```

版本	修改内容	修改人
v1.0.0	init	刘章术

1.组件说明

1. MetricServer：是kubernetes集群资源使用情况的聚合器，收集数据给kubernetes集群内使用，如kubectl,hpa,scheduler等。
2. PrometheusOperator：是一个系统监测和警报工具箱，用来存储监控数据。
3. NodeExporter：用于各node的关键度量指标状态数据。
4. KubeStateMetrics：收集kubernetes集群内资源对象数据，制定告警规则。
5. Prometheus：采用pull方式收集apiserver，scheduler，controller-manager，kubelet组件数据，通过http协议传输。
6. Grafana：是可视化数据统计和监控平台

2.构建

1. 拉取工程

```
git clone https://github.com/coreos/kube-prometheus.git
cd kube-prometheus/manifests
```

2. 镜像导入

```
docker load -i  addon-resizer.tar
docker load -i alertmanager.tar
docker load -i configmap-reload.tar
docker load -i k8s-prometheus-adapter-amd64.tar
docker load -i kube-rbac-proxy.tar
docker load -i kube-state-metrics.tar
docker load -i node-exporter.tar
docker load -i prometheus-config-reloader.tar
docker load -i prometheus-operator.tar
docker load -i prometheus.tar
docker load -i grafana.tar
```

3. 修改grafana-service.yaml 文件，使用 nodepode 方式访问 grafana：

```
apiVersion: v1
kind: Service
metadata:
  name: grafana
  namespace: monitoring
spec:
  type: NodePort #添加内容
  ports:
  - name: http
    port: 3000
    targetPort: http
    nodePort: 30100  #添加内容
  selector:
    app: grafana
```

4. 修改prometheus-service.yaml，改为 nodepode

```
apiVersion: v1
kind: Service
metadata:
  labels:
  prometheus: k8s
  name: prometheus-k8s
  namespace: monitoring
spec:
  type: NodePort
  ports:
  - name: web
    port: 9090
    targetPort: web
    nodePort: 30200
  selector:
    app: prometheus
    prometheus: k8s
```

5. 修改alertmanager-service.yaml，改为 nodepode

```
apiVersion: v1
kind: Service
metadata:
  labels:
    alertmanager: main
  name: alertmanager-main
  namespace: monitoring
spec:
  type: NodePort
  ports:
  - name: web
    port: 9093
    targetPort: web
    nodePort: 30300
  selector:
    alertmanager: main
    app: alertmanager
```

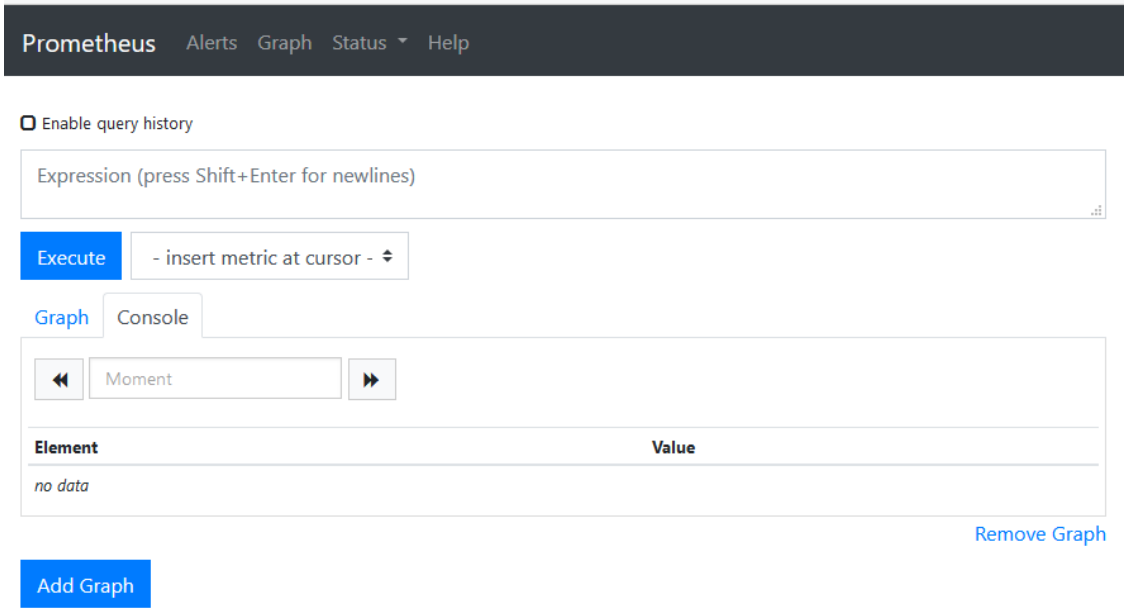
```
  sessionAffinity: ClientIP
```

6. 安装资源

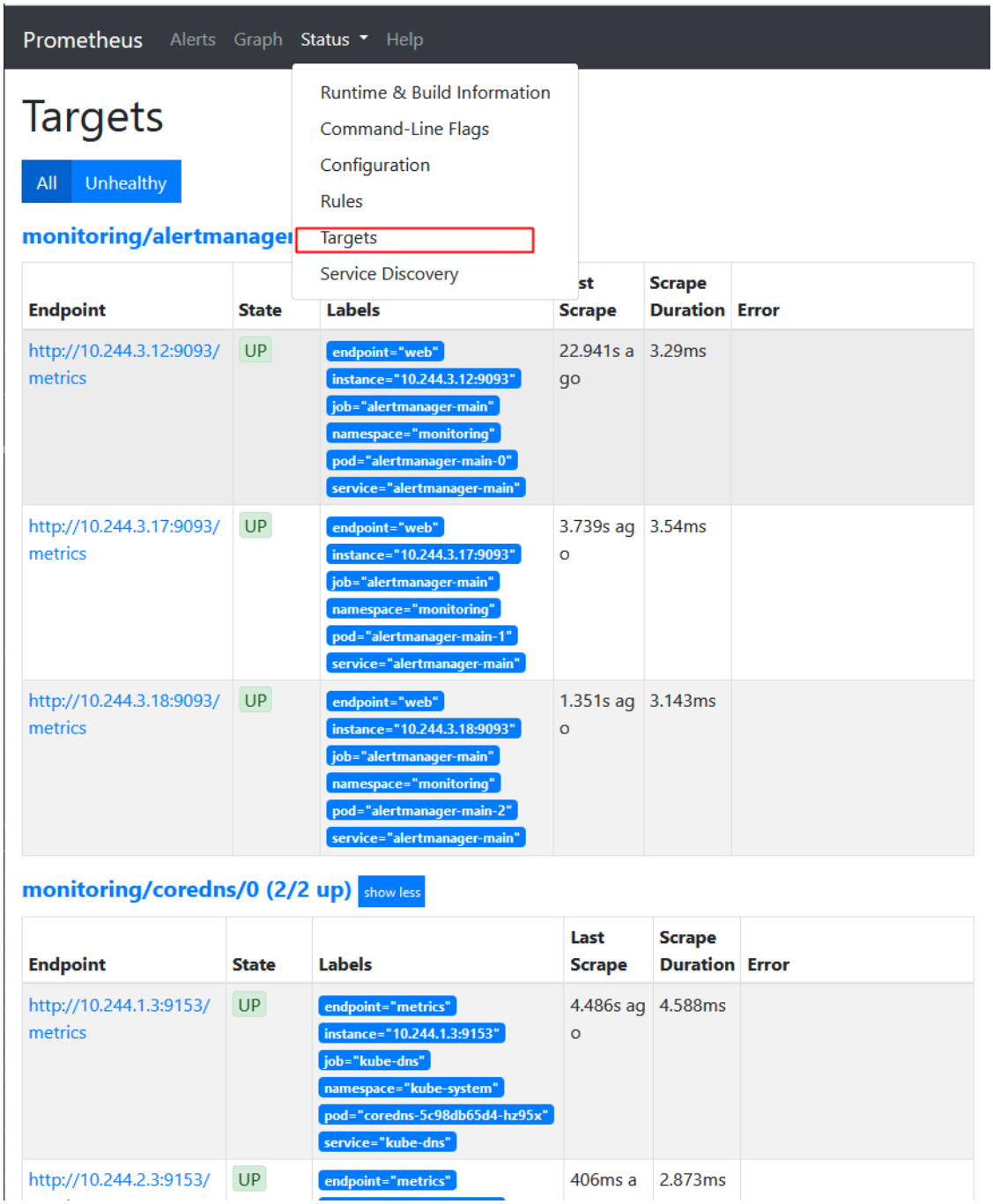
```
# 需要互相链接，需要多允许几次
kubectl apply -f ../manifests/
```

7. 访问promethues

promethues对应的nodeport端口为30200访问 http://masterip:30200



通过访问http://MasterIP:30200/target 可以看到 prometheus 已经成功连接上了 k8s 的 apiserver



查看service-discovery

Prometheus 自己的指标:http://master1:30200//metrics

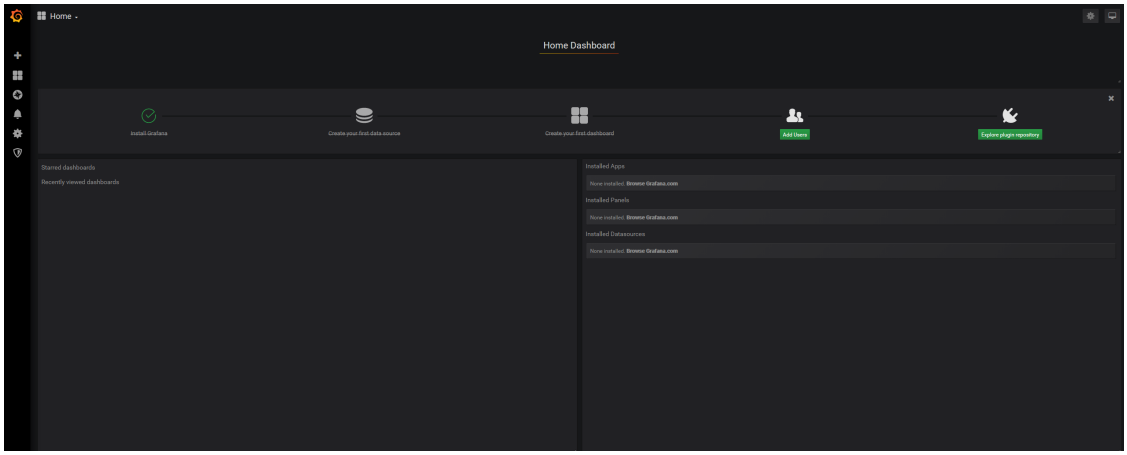
```
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 1.2593e-05
go_gc_duration_seconds{quantile="0.25"} 2.118e-05
go_gc_duration_seconds{quantile="0.5"} 3.1189e-05
go_gc_duration_seconds{quantile="0.75"} 4.5556e-05
go_gc_duration_seconds{quantile="1"} 0.003453028
go_gc_duration_seconds_sum 0.00833808
go_gc_duration_seconds_count 80
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 248
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.12.7"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.44941032e+08
# 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 8.775255016e+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 1.856686e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 5.216412e+07
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 0.003894280850581415
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 1.348608e+07
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 1.44941032e+08
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
```

prometheus 的 WEB 界面上提供了基本的查询 K8S 集群中每个 POD 的 CPU 使用情况(使用 PromQL)，查询条件如下

```
sum by (pod_name)( rate(container_cpu_usage_seconds_total{image!="",
pod_name!=""}[1m] ))
```

8. 访问grafana

浏览器访问 http://MasterIP:30100 用户名密码默认 admin/admin



附录

Prometheus github 地址：https://github.com/coreos/kube-prometheus