k8s 部署Prometheus

简介

本文档会指导你如何在kubernetes集群上部署prometheus并使用其监控你的集群状态，prometheus是高度可扩展的开源架构，它提供了对kubernetes开箱即用的监控功能，解析prometheus被分为了几个部分，如果你想了解更多内容，可以参考https://prometheus.io/

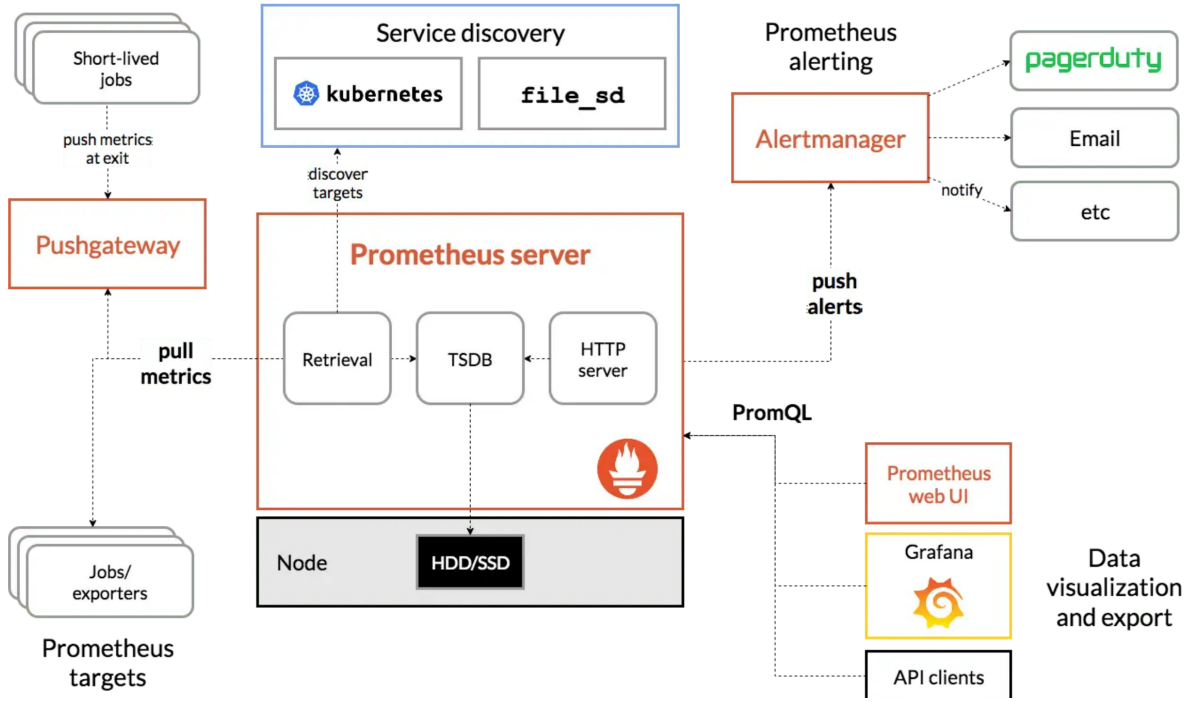
1 **Metric collection：**Prometheus 使用http协议pull 模式拉取监控指标，对于无法直接拉取的指标，prometheus采用push-gateway 直接推送监控数据到prometheus监控端

2 **Metric Endpoint：** 对于提供监控数据的服务，通常使用/metrics 暴露监控数据给server端定期拉取

3 **PromQL：**Prometheus通过PromQL查询数据，你可以直接在Prometheus的监控面板上直接执行语句查询指标，或者通过grafana调整PromQL图形化展示数据

4 **Prometheus Exporters**：Exporter库转换来自第三方的服务指标，使其可以通过PromQL语句抓取，官方提供了很多的Exporter库，比较常见的Exporter使用方式就是，node-Exporter，提取os层的服务指标

5 **TSDB**： Prometheus 使用可以有效的使用TSDB存储监控数据，默认的方式是直接存储在本地的文件系统内，但是你也可以修改配置，将监控指标存储在Prometheus TSDB中



本文档架设你已经部署了kubernetes集群，并且熟悉kubectl，如果你还没有配置过kubernetes集群，你可以通过这个链接快速部署一套minikube https://devopscube.com/kubernetes-minikube-tutorial/

本教程提供的所有yaml文件都已经放置在github，你可以通过clone的方式获取

```
Plain Text
1  git clone https://github.com/lvtujingji/kubernetes-prometheus.git
```

1 部署Prometheus-server

1.1 创建名称空间和ROLE

```
Plain Text
1  kubectl create namespace monitoring
```

Prometheus使用kubernetes API获取node 、pod、deploymend的Metric，因此我们必须对通过RBAC的方式对Prometheus进行授权，需要注意的是这个role绑定在monitoring下的default

ServiceAccount，如果你使用的是其他SA，需要调整你的绑定用户

Plain Text

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: prometheus
rules:
- apiGroups: [""]
  resources:
  - nodes
  - nodes/proxy
  - services
  - endpoints
  - pods
  verbs: ["get", "list", "watch"]
- apiGroups:
  - extensions
  resources:
  - ingresses
  verbs: ["get", "list", "watch"]
- nonResourceURLs: ["/metrics"]
  verbs: ["get"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: prometheus
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: prometheus
subjects:
- kind: ServiceAccount
  name: default
  namespace: monitoring
```

1.2 创建Prometheus外部配置文件

通过configmap去调整服务的配置文件，你不需要每次修改配置文件的时候都重新构建镜像，只需要重新应用下configmap，然后重启下pod就可以生效，这个configmap将配置文件挂载在/etc/prometheus路径下，分别被命名为 prometheus.yaml 和 prometheus.rules

```
kubectl create -f config-map.yaml
```

所有的监控报警规则被配置在prometheus.rules文件内这个会在altermanager部分介绍

所有的发现信息和存储信息被配置在prometheus.yaml文件内,包含了pod和service的动态发现

Plain Text

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
  labels:
    name: prometheus-config
  namespace: monitoring
data:
  prometheus.rules: |-
    groups:
    - name: devopscube demo alert
      rules:
      - alert: PrometheusJobMissing
        expr: absent(up{kubernetes_name="prometheus-service"})
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Prometheus job missing (instance {{ $labels.instance }})
          description: "A Prometheus job has disappeared\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: PrometheusTargetMissing
        expr: up == 0
        for: 0m
        labels:
          severity: critical
        annotations:
          summary: Prometheus target missing (instance {{ $labels.instance }})
          description: "A Prometheus target has disappeared. An exporter might be crashed.\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: PrometheusTooManyRestarts
        expr: changes({kubernetes_name=~"prometheus|alertmanager|node-exporter|grafana"}[15m]) > 2
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Prometheus too many restarts (instance {{ $labels.instance }})
          description: "Prometheus has restarted more than twice in the last 15 minutes. It might be crashlooping.\n  VALUE = {{ $value }}\n  LABE
```

```
LS = {{ $labels }}"

      - alert: HostOutOfMemory
        expr: (node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes * 100 < 10) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: Host out of memory (instance {{ $labels.instance }})
          description: "Node memory is filling up (< 10% left)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: HostMemoryUnderMemoryPressure
        expr: (rate(node_vmstat_pgmajfault[1m]) > 1000) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: Host memory under memory pressure (instance {{ $labels.instance }})
          description: "The node is under heavy memory pressure. High rate of major page faults\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: HostUnusualNetworkThroughputIn
        expr: (sum by (instance) (rate(node_network_receive_bytes_total[2m])) / 1024 / 1024 > 100) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: Host unusual network throughput in (instance {{ $labels.instance }})
          description: "Host network interfaces are probably receiving too much data (> 100 MB/s)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: HostUnusualNetworkThroughputOut
        expr: (sum by (instance) (rate(node_network_transmit_bytes_total[2m])) / 1024 / 1024 > 100) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 5m
        labels:
          severity: warning
        annotations:
```

```
          summary: Host unusual network throughput out (instance {{ $labels.instance }})
          description: "Host network interfaces are probably sending too much data (> 100 MB/s)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: HostUnusualDiskReadRate
        expr: (sum by (instance) (rate(node_disk_read_bytes_total[2m])) / 1024 / 1024 > 50) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: Host unusual disk read rate (instance {{ $labels.instance }})
          description: "Disk is probably reading too much data (> 50 MB/s)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: HostUnusualDiskWriteRate
        expr: (sum by (instance) (rate(node_disk_written_bytes_total[2m])) / 1024 / 1024 > 50) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: Host unusual disk write rate (instance {{ $labels.instance }})
          description: "Disk is probably writing too much data (> 50 MB/s)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: HostOutOfDiskSpace
        expr: ((node_filesystem_avail_bytes * 100) / node_filesystem_size_bytes < 10 and ON (instance, device, mountpoint) node_filesystem_readonly == 0) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: Host out of disk space (instance {{ $labels.instance }})
          description: "Disk is almost full (< 10% left)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: HostHighCpuLoad
        expr: (sum by (instance) (avg by (mode, instance) (rate(node_cpu_seconds_total{mode!="idle"}[2m]))) > 0.85) * on(instance) group_left (nod
```

```
ename) node_uname_info{nodename=~".+"}
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: Host high CPU load (instance {{ $labels.instance }})
          description: "CPU load is > 80%\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: HostOomKillDetected
        expr: (increase(node_vmstat_oom_kill[1m]) > 0) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Host OOM kill detected (instance {{ $labels.instance }})
          description: "OOM kill detected\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesNodeNotReady
        expr: kube_node_status_condition{condition="Ready",status="true"} == 0
        for: 10m
        labels:
          severity: critical
        annotations:
          summary: Kubernetes Node not ready (instance {{ $labels.instance }})
          description: "Node {{ $labels.node }} has been unready for a long time\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesNodeMemoryPressure
        expr: kube_node_status_condition{condition="MemoryPressure",status="true"} == 1
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: Kubernetes Node memory pressure (instance {{ $labels.instance }})
          description: "Node {{ $labels.node }} has MemoryPressure condition\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesNodeNetworkUnavailable
        expr: kube_node_status_condition{condition="NetworkUnavailable",status="true"} == 1
```

```
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: Kubernetes Node network unavailable (instance {{ $labels.instance }})
          description: "Node {{ $labels.node }} has NetworkUnavailable condition\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesContainerOomKiller
        expr: (kube_pod_container_status_restarts_total - kube_pod_container_status_restarts_total offset 10m >= 1) and ignoring (reason) min_over_time(kube_pod_container_status_last_terminated_reason{reason="OOMKilled"}[10m]) == 1
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes Container oom killer (instance {{ $labels.instance }})
          description: "Container {{ $labels.container }} in pod {{ $labels.namespace }}/{{ $labels.pod }} has been OOMKilled {{ $value }} times in the last 10 minutes.\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesJobFailed
        expr: kube_job_status_failed > 0
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes Job failed (instance {{ $labels.instance }})
          description: "Job {{ $labels.namespace }}/{{ $labels.job_name }} failed to complete\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesCronjobSuspended
        expr: kube_cronjob_spec_suspend != 0
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes CronJob suspended (instance {{ $labels.instance }})
          description: "CronJob {{ $labels.namespace }}/{{ $labels.cronjob }} is suspended\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesPersistentvolumeclaimPending
```

```
        expr: kube_persistentvolumeclaim_status_phase{phase="Pending"} == 1
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes PersistentVolumeClaim pending (instance {{ $labels.instance }})
          description: "PersistentVolumeClaim {{ $labels.namespace }}/{{ $labels.persistentvolumeclaim }} is pending\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesVolumeOutOfDiskSpace
        expr: kubelet_volume_stats_available_bytes / kubelet_volume_stats_capacity_bytes * 100 < 10
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes Volume out of disk space (instance {{ $labels.instance }})
          description: "Volume is almost full (< 10% left)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesPersistentvolumeError
        expr: kube_persistentvolume_status_phase{phase=~"Failed|Pending", job="kube-state-metrics"} > 0
        for: 0m
        labels:
          severity: critical
        annotations:
          summary: Kubernetes PersistentVolume error (instance {{ $labels.instance }})
          description: "Persistent volume {{ $labels.persistentvolume }} is in bad state\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesStatefulsetDown
        expr: kube_statefulset_replicas != kube_statefulset_status_replicas_ready > 0
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: Kubernetes StatefulSet down (instance {{ $labels.instance }})
          description: "StatefulSet {{ $labels.namespace }}/{{ $labels.statefulset }} went down\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
```

```
      - alert: KubernetesApiServerErrors
        expr: sum(rate(apiserver_request_total{job="kubernetes-apiservers",code=~"^(?:5..)$"}[1m])) / sum(rate(apiserver_request_total{job="apiserver"}[1m])) * 100 > 3
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: Kubernetes API server errors (instance {{ $labels.instance }})
          description: "Kubernetes API server is experiencing high error rate\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

      - alert: KubernetesClientCertificateExpiresNextWeek
        expr: apiserver_client_certificate_expiration_seconds_count{job="kubernetes-apiservers"} > 0 and histogram_quantile(0.01, sum by (job, le) (rate(apiserver_client_certificate_expiration_seconds_bucket{job="kubernetes-apiservers"}[5m]))) < 7*24*60*60
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes client certificate expires next week (instance {{ $labels.instance }})
          description: "A client certificate used to authenticate to the apiserver is expiring next week.\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"

  prometheus.yml: |-
    global:
      scrape_interval: 5s
      evaluation_interval: 5s
    rule_files:
      - /etc/prometheus/prometheus.rules
    # 监控配置
    alerting:
      alertmanagers:
      - scheme: http
        static_configs:
        - targets:
          - "alertmanager.monitoring.svc:9093"
    # 动态发现配置
    scrape_configs:
      - job_name: 'kubernetes-apiservers'
        # 大致意思是说，筛选__meta_kubernetes_namespace，__meta_kubernetes_service_name，__meta_kubernetes_endpoint_port_name
        # 为default，kubernetes，https，也就是说发现文件的路径为https://kubernetes.default:$port/metircs
```

```
        # 符合这个条件的只有kubernetes 这个endpoint，所以只会发现apiserver
        kubernetes_sd_configs:
        - role: endpoints
        scheme: https
        tls_config:
          ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
        relabel_configs:
        - source_labels: [__meta_kubernetes_namespace, __meta_kubernetes_service_name, __meta_kubernetes_endpoint_port_name]
          action: keep
          regex: default;kubernetes;https
      - job_name: 'node-exporter'
        # 同理这个匹配规则是匹配__meta_kubernetes_endpoints_name 值为node-exporter的目标
        kubernetes_sd_configs:
        - role: endpoints
        relabel_configs:
        - source_labels: [__meta_kubernetes_endpoints_name]
          regex: 'node-exporter'
          action: keep
      - job_name: 'kubernetes-nodes'
        #将抓取的地址调整为https://kubernetes.default.svc:443/api/v1/nodes/${node_name}/proxy/metrics
        scheme: https
        tls_config:
          ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
        kubernetes_sd_configs:
        - role: node
        relabel_configs:
        - action: labelmap
          regex: __meta_kubernetes_node_label_(.+)
        - target_label: __address__
          replacement: kubernetes.default.svc:443
        - source_labels: [__meta_kubernetes_node_name]
          regex: (.+)
          target_label: __metrics_path__
          replacement: /api/v1/nodes/${1}/proxy/metrics
      - job_name: 'kubernetes-pods'
        # 只抓取annotation中 prometheus_io_scrape: true的pod
        # 将抓取的路径/metircs 调整为prometheus_io_path 设置的值
        # 定义addres抓取端口
        # labelmap的意思是将kubernetes_pod_label 所有标签都映射到prometheus target面板
```

```
        # 然后自定义俩个label分别是kubernetes_namespace 对应值__meta_kubernetes_namespace
        kubernetes_sd_configs:
        - role: pod
        relabel_configs:
        - source_labels: [__meta_kubernetes_pod_annotation_prometheus_io_scrape]
          action: keep
          regex: true
        - source_labels: [__meta_kubernetes_pod_annotation_prometheus_io_path]
          action: replace
          target_label: __metrics_path__
          regex: (.+)
        - source_labels: [__address__, __meta_kubernetes_pod_annotation_prometheus_io_port]
          action: replace
          regex: ([^:]+)(?::\d+)?;(\d+)
          replacement: $1:$2
          target_label: __address__
        - action: labelmap
          regex: __meta_kubernetes_pod_label_(.+)
        - source_labels: [__meta_kubernetes_namespace]
          action: replace
          target_label: kubernetes_namespace
        - source_labels: [__meta_kubernetes_pod_name]
          action: replace
          target_label: kubernetes_pod_name
      - job_name: 'kube-state-metrics'
       #直接指定的静态路径
        static_configs:
          - targets: ['kube-state-metrics.kube-system.svc:8080']
      - job_name: 'kubernetes-cadvisor'
        # 同上
        scheme: https
        tls_config:
          ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
        kubernetes_sd_configs:
        - role: node
        relabel_configs:
        - action: labelmap
          regex: __meta_kubernetes_node_label_(.+)
        - target_label: __address__
          replacement: kubernetes.default.svc:443
        - source_labels: [__meta_kubernetes_node_name]
          regex: (.+)
```

```
          target_label: __metrics_path__
          replacement: /api/v1/nodes/${1}/proxy/metrics/cadvisor
      - job_name: 'kubernetes-service-endpoints'
        kubernetes_sd_configs:
        - role: endpoints
        relabel_configs:
        - source_labels: [__meta_kubernetes_service_annotation_prometheus
_io_scrape]
          action: keep
          regex: true
        - source_labels: [__meta_kubernetes_service_annotation_prometheus
_io_scheme]
          action: replace
          target_label: __scheme__
          regex: (https?)
        - source_labels: [__meta_kubernetes_service_annotation_prometheus
_io_path]
          action: replace
          target_label: __metrics_path__
          regex: (.+)
        - source_labels: [__address__, __meta_kubernetes_service_annotati
on_prometheus_io_port]
          action: replace
          target_label: __address__
          regex: ([^:]+)(?::\d+)?;(\d+)
          replacement: $1:$2
        - action: labelmap
          regex: __meta_kubernetes_service_label_(.+)
        - source_labels: [__meta_kubernetes_namespace]
          action: replace
          target_label: kubernetes_namespace
        - source_labels: [__meta_kubernetes_service_name]
          action: replace
          target_label: kubernetes_name
```

1.3 创建prometheus deployment

使用deployment.yaml 部署服务，需要注意的是这个deployment使用的是最新的prometheus镜像，另外并未使用持久盘，如果是生产环境部署，请提前创建持久盘，并修改deployment卷部分

Plain Text

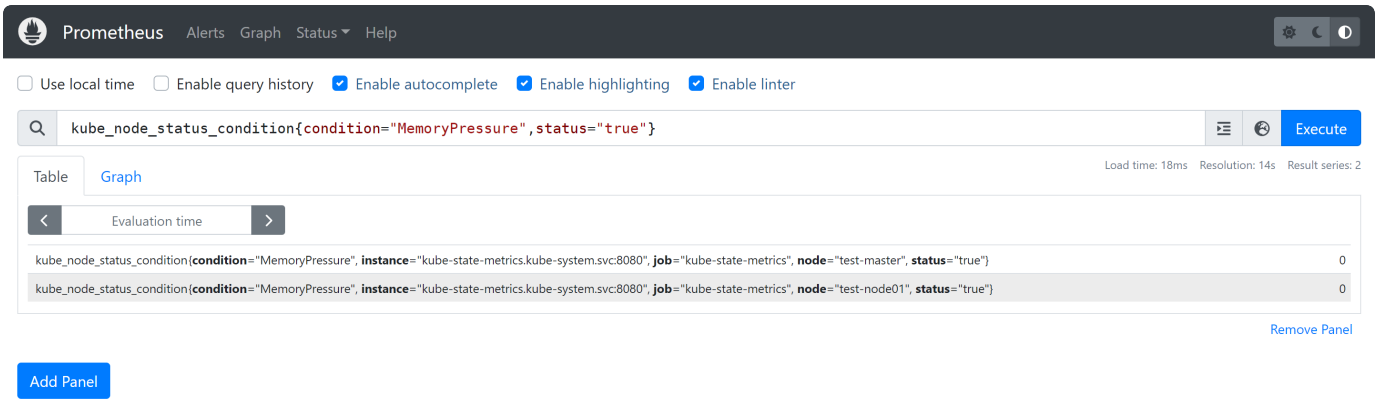
```
kubectl apply -f deployment.yaml
```

1.4 通过Endpoint 对外提供服务

如果需要访问prometheus的控制面板，需要通过kubernetes service服务的方式对外开放端口

```
Plain Text
1  kubetl apply -f service.yaml
```

当这个服务创建后，可以通过访问nodeip 30000端口的方式访问到，prometheus 端点



2 部署exporter

在安装node-exporter 和kube-state-metrics 之前有必要先说明二者的区别

node-exporter 关注的是底层资源， 主要是监控 CPU 使用率、内存使用率、磁盘空间、网络流量等

kube-state-metrics 负责的关注点是 Kubernetes 集群的状态，state-metrics通过访问 kubernetes api 获取大量的监控指标如 Pod 运行状态、副本集状态、服务状态等，然后将其加工为prometheus可以抓取的数据类型。

2.1 创建node-exporte配置文件

此 Daemonset 将部署在监控命名空间中。如果您希望将其部署在不同的命名空间中，请在以下 YAML 中进行更改，该目录下包含了demonset 和service 应用

```
Plain Text
1  cd nodeExporter
2  kubectl apply -f ./
```

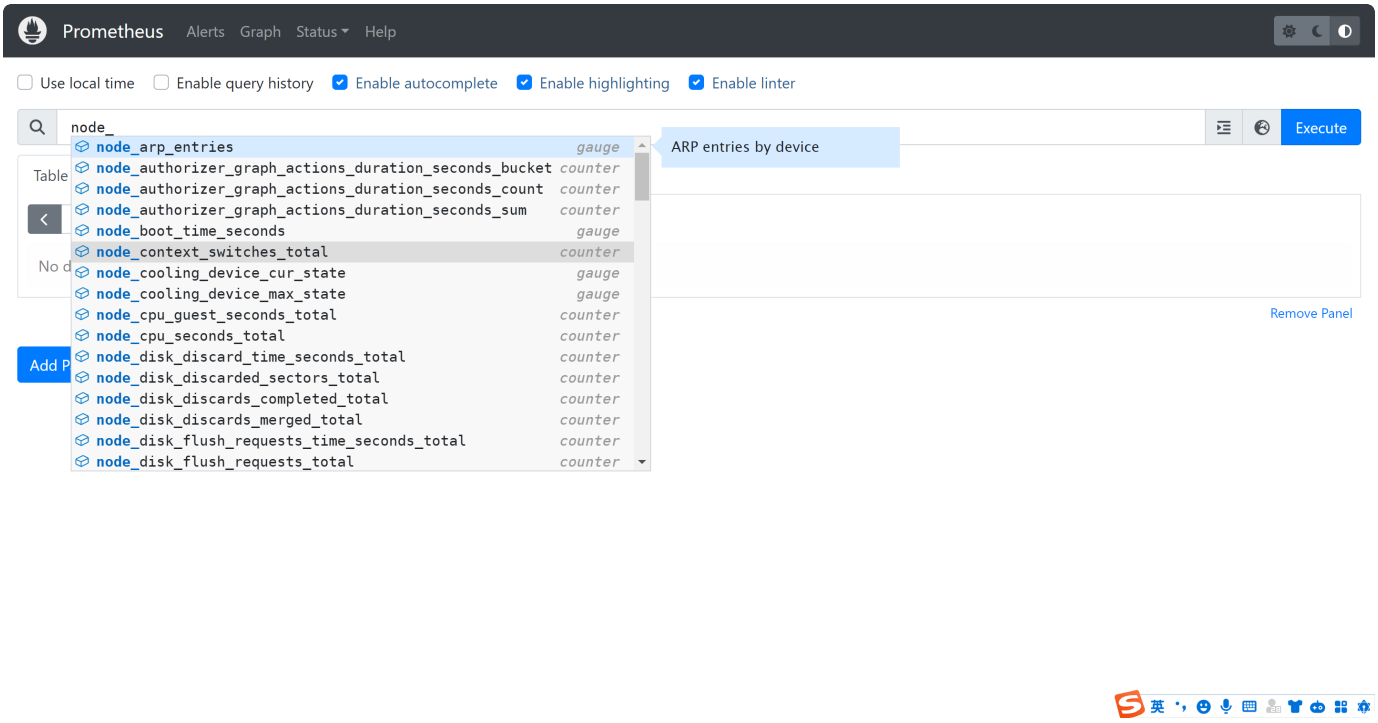
2.2 查看应用状态

```
# 列出daemonset监控命名空间中的并确保其处于可用状态。
kubectl get daemonset -n monitoring
kubectl get endpoints -n monitoring

# 通过Endpoint可以看到svc已经关联到pod的相应端口上
#NAMESPACE      NAME                              ENDPOINTS
#monitoring     node-exporter                     10.244.130.102:9100
```

2.3 查看node-exporter 监控指标

来自节点导出器的所有指标都带有前缀`node_`

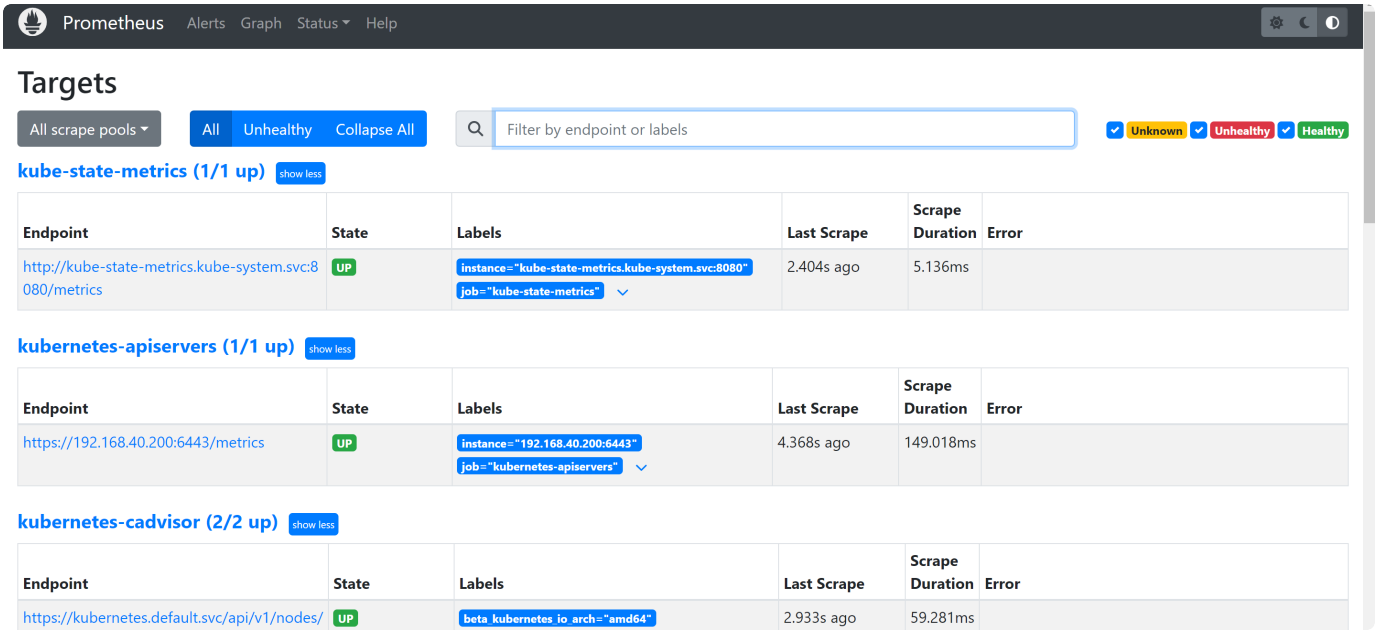


2.4 部署kube-state-metrics

所有 Kube 静态指标都可以从 URI 上的 Kube 状态服务端点获取`/metrics`。

此配置可以添加为**Prometheus 作业配置**的一部分。您需要将以下作业配置添加到 Prometheus 配置中，以便 Prometheus 抓取所有 Kube 状态指标。

```
cd stateMetrics/
kubectl apply -f ./
```



3 部署altermanager

alter–manager 是开源的告警系统，它接收来自Prometheus的所有告警信息，然后将这些告警推送到指定的收件人列表中，收件人可以是邮箱，钉钉或者其他web–hook，安装步骤分为下列4个部分

1 修改配置告警收件人，告警模板

2 配置deployment

3 配置钉钉webhook–deployment

4 prometheus.rules 告警规则

3.1 修改配置告警收件人

在开始配置之前，请确保你的Prometheus已经部署且正运行，确保prometheus 配置文件中告警部分配置为下列内容

```
alerting:
   alertmanagers:
      - scheme: http
        static_configs:
        - targets:
          - "alertmanager.monitoring.svc:9093"
```

网易的发件secret 获取

登录到网易邮箱，点击红框部分确保pop3/SMTP状态为开启，然后点击新增授权密码，可能需要短信认证

alterManager目录里记载了所有的alter相关配置文件

Plain Text

```
cd alertManager/
cat alertManagerConfigmap.yaml

kind: ConfigMap
apiVersion: v1
metadata:
  name: alertmanager-config
  namespace: monitoring
data:
  config.yml: |-
    global:
      # 发件人，和发件凭证注意是凭证passwor对应之前创建的secret，不是登录密码这些都需要修改
      smtp_smarthost: 'smtp.163.com:465'
      smtp_from: lvtujingji@163.com
      smtp_auth_username: lvtujingji@163.com
      smtp_auth_password: ################
      smtp_require_tls: false
    templates:
    - '/etc/alertmanager/*.tmpl'
    route:
      # 收件人name,group_by 是基于alertname priority 分组
      # group_wait 组合并等待时间 repeat_interval重发送时间
      receiver: alert-emailer
      group_by: ['alertname', 'priority']
      group_wait: 10s
      repeat_interval: 1m
        # 如果需要多联系人过滤，就可以修改这部分，匹配severity=slack的告警，然后发送到收件人slack_demo
        #routes:
        #- receiver: slack_demo
        # Send severity=slack alerts to slack.
        #  match:
        #    severity: slack
        #  group_wait: 10s
        #  repeat_interval: 1m
    # 收件人信息，可以配置邮箱地址，我需要发送到钉钉，所以填的webhook地址
    receivers:
    - name: alert-emailer
      webhook_configs:
      - url: 'http://prometheus-webhook-dingtalk.monitoring.svc/dingtalk/webhook_ding/send'
        send_resolved: true
  # email addres example
   #- name: email-notifications
```

```
43        # email_configs:
44         #  - to: your-email@example.com
45
```

3.2 配置deployment

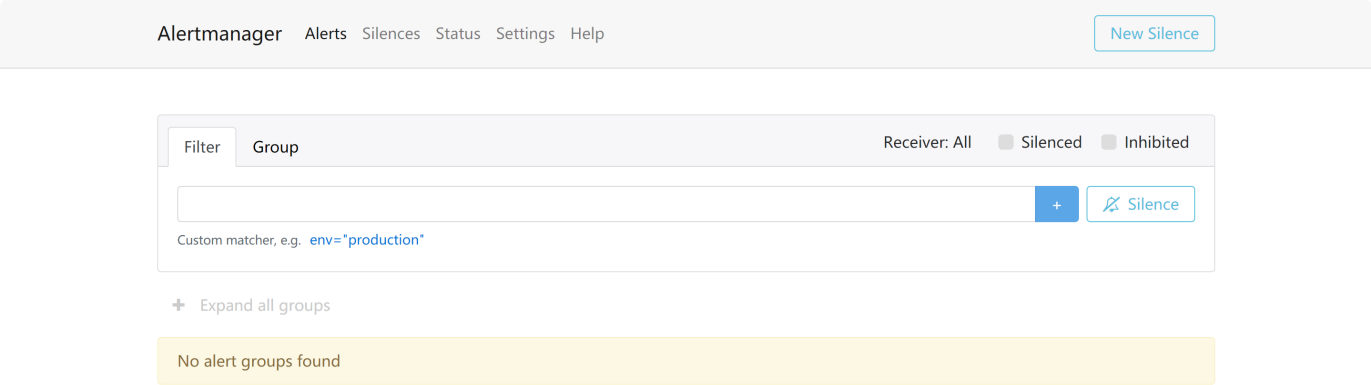
执行下列代码之前确保当前路径在alertManager/

```
Plain Text
1   kubectl apply -f ./
```

apply 会在monitor名称空间下创建altermanager deployment和serivce，对外的开放的nodeport是31000，通过get命令获取到service状态信息

```
Plain Text
1    kubectl get -n monitoring svc
2   NAME                         TYPE        CLUSTER-IP       EXTERNAL-IP   PO
    RT(S)          AGE
3   alertmanager                 NodePort    10.110.212.137   <none>        90
    93:31000/TCP   4d6h
4   node-exporter                ClusterIP   10.102.77.171    <none>        91
    00/TCP         4d6h
5   prometheus-service           NodePort    10.102.142.208   <none>        80
    80:30000/TCP   4d7h
6
```

输入ip+31000端口可以进入altermanager控制面板，可以看到没有任何告警信息



3.3 配置钉钉webhook-deployment(可选)

如果要发送告警信息到钉钉，就需要配置这部分，其他webhook应用原理大概都一致，都是创建个告警机器人，然后记录告警地址，修改alertWebhook/configMap.yaml

测试的话可以自己创建个群聊，然后点击机器人

记录webhook和加签里的secret

Plain Text

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-webhook-dingtalk
  namespace: monitoring
data:
  config.yml: |
    targets:
    # 将 URL替换成你钉钉告警机器的人的地址
    # 将secre替换成你钉钉的secret
      webhook_ding:
        url: https://oapi.dingtalk.com/robot/send?access_token=1dfa9f2d57649ace8ecd5917814a8c8d43201419f421ba5155f02936d8aec51b  # 配置机器人的webhook_url)
        secret: SECd2239c2db306ad97e800fbab4fdc9b1c8df58d368c0eb8fda1f7f3e81e3312a6
        message:
          title: '{{ template "default.title" . }}'
          text: '{{ template "default.content" . }}'
```

修改完毕之后直接应用就行

```
kubectl apply -f ./
[root@test-master alertWebhook]# kubectl get svc -n monitoring
NAME                          TYPE        CLUSTER-IP       EXTERNAL-IP   PO
RT(S)          AGE
alertmanager                  NodePort    10.110.212.137   <none>        90
93:31000/TCP   4d6h
node-exporter                 ClusterIP   10.102.77.171    <none>        91
00/TCP         4d6h
prometheus-service            NodePort    10.102.142.208   <none>        80
80:30000/TCP   4d7h
prometheus-webhook-dingtalk   ClusterIP   10.96.200.178    <none>        8
0/TCP           4d6h

```

3.4 prometheus.rules 告警规则

这部分是关于Prometheus监控告警规则的大致介绍，大部分都是直接从下面那个地址获取的，我截取了我目前需要的一部分，主要分为basic和kubernetes俩个部分的监控告警，也可以根据你们的应用情况，去截取修改

https://samber.github.io/awesome-prometheus-alerts/

Plain Text

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
  labels:
    name: prometheus-config
  namespace: monitoring
data:
  prometheus.rules: |-
    groups:
    # 可以基于group的name进行监控项分类，然后进行告警收件人分类
    - name: basic
      rules:
        # 这个监控的意思是当获取up{kubernetes_name="prometheus-service"} 抓取不到任何监控时
        # 判断这个Prometheus-service已经出现问题了，然后触发告警
      - alert: PrometheusJobMissing
        expr: absent(up{kubernetes_name="prometheus-service"})
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Prometheus job missing (instance {{ $labels.instance }})
          description: "A Prometheus job has disappeared\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
        # 如果出现自动发现的目标断开就会触发告警
      - alert: PrometheusTargetMissing
        expr: up == 0
        for: 0m
        labels:
          severity: critical
        annotations:
          summary: Prometheus target missing (instance {{ $labels.instance }})
          description: "A Prometheus target has disappeared. An exporter might be crashed.\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
        #统计Prometheus关键组件有没有出现频繁重启现象
      - alert: PrometheusTooManyRestarts
        expr: changes(process_start_time_seconds{kubernetes_name=~"prometheus-service|alertmanager|node-exporter"}[15m]) > 2
        for: 0m
        labels:
          severity: warning
        annotations:
```

```
40            summary: Prometheus too many restarts (instance {{ $labels.inst
    ance }})
41            description: "Prometheus has restarted more than twice in the l
    ast 15 minutes. It might be crashlooping.\n  VALUE = {{ $value }}\n  LABE
    LS = {{ $labels }}"
42          # 内存的可用率小于10%
43        - alert: HostOutOfMemory
44          expr: node_memory_MemAvailable_bytes / node_memory_MemTotal_byte
    s * 100 < 10
45          for: 2m
46          labels:
47            severity: warning
48          annotations:
49            summary: Host out of memory (instance {{ $labels.instance }})
50            description: "Node memory is filling up (< 10% left)\n  VALUE
    = {{ $value }}\n  LABELS = {{ $labels }}"
51          # 网络传入流量超过100M/s
52        - alert: HostUnusualNetworkThroughputIn
53          expr: sum by (instance) (rate(node_network_receive_bytes_total[2
    m])) / 1024 / 1024 > 100
54          for: 5m
55          labels:
56            severity: warning
57          annotations:
58            summary: Host unusual network throughput in (instance {{ $label
    s.instance }})
59            description: "Host network interfaces are probably receiving to
    o much data (> 100 MB/s)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels
    }}"
60        #网络传出流量超过100M/s
61        - alert: HostUnusualNetworkThroughputOut
62          expr: sum by (instance) (rate(node_network_transmit_bytes_total[2
    m])) / 1024 / 1024 > 100
63          for: 5m
64          labels:
65            severity: warning
66          annotations:
67            summary: Host unusual network throughput out (instance {{ $labe
    ls.instance }})
68            description: "Host network interfaces are probably sending too
    much data (> 100 MB/s)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
69        # 磁盘读取速率超出50MB/s
70        - alert: HostUnusualDiskReadRate
71          expr: (sum by (instance) (rate(node_disk_read_bytes_total[2m]))
    / 1024 / 1024 > 50) * on(instance) group_left (nodename) node_uname_info
    {nodename=~".+"}
72          for: 5m
73          labels:
```

```
          severity: warning
        annotations:
          summary: Host unusual disk read rate (instance {{ $labels.instance }})
          description: "Disk is probably reading too much data (> 50 MB/s)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
       # 磁盘写入速率超出50MB/s
      - alert: HostUnusualDiskWriteRate
        expr: (sum by (instance) (rate(node_disk_written_bytes_total[2m])) / 1024 / 1024 > 50) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: Host unusual disk write rate (instance {{ $labels.instance }})
          description: "Disk is probably writing too much data (> 50 MB/s)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # 磁盘剩余空间小于10%
      - alert: HostOutOfDiskSpace
        expr: ((node_filesystem_avail_bytes * 100) / node_filesystem_size_bytes < 10 and ON (instance, device, mountpoint) node_filesystem_readonly == 0) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: Host out of disk space (instance {{ $labels.instance }})
          description: "Disk is almost full (< 10% left)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # CPU 10分钟的负载超过80%
      - alert: HostHighCpuLoad
        expr: (sum by (instance) (avg by (mode, instance) (rate(node_cpu_seconds_total{mode!="idle"}[2m]))) > 0.85) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 10m
        labels:
          severity: warning
        annotations:
          summary: Host high CPU load (instance {{ $labels.instance }})
          description: "CPU load is > 80%\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # 主机内出现了OOM
      - alert: HostOomKillDetected
```

```
        expr: (increase(node_vmstat_oom_kill[1m]) > 0) * on(instance) group_left (nodename) node_uname_info{nodename=~".+"}
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Host OOM kill detected (instance {{ $labels.instance }})
          description: "OOM kill detected\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # node状态notREADY
      - alert: KubernetesNodeNotReady
        expr: kube_node_status_condition{condition="Ready",status="true"} == 0
        for: 10m
        labels:
          severity: critical
        annotations:
          summary: Kubernetes Node not ready (instance {{ $labels.instance }})
          description: "Node {{ $labels.node }} has been unready for a long time\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # node面临内存压力
      - alert: KubernetesNodeMemoryPressure
        expr: kube_node_status_condition{condition="MemoryPressure",status="true"} == 1
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: Kubernetes Node memory pressure (instance {{ $labels.instance }})
          description: "Node {{ $labels.node }} has MemoryPressure condition\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # node网络出现不可用现象
      - alert: KubernetesNodeNetworkUnavailable
        expr: kube_node_status_condition{condition="NetworkUnavailable",status="true"} == 1
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: Kubernetes Node network unavailable (instance {{ $labels.instance }})
          description: "Node {{ $labels.node }} has NetworkUnavailable condition\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # nodekill 容器，kill原因内存不足或超出
      - alert: KubernetesContainerOomKiller
```

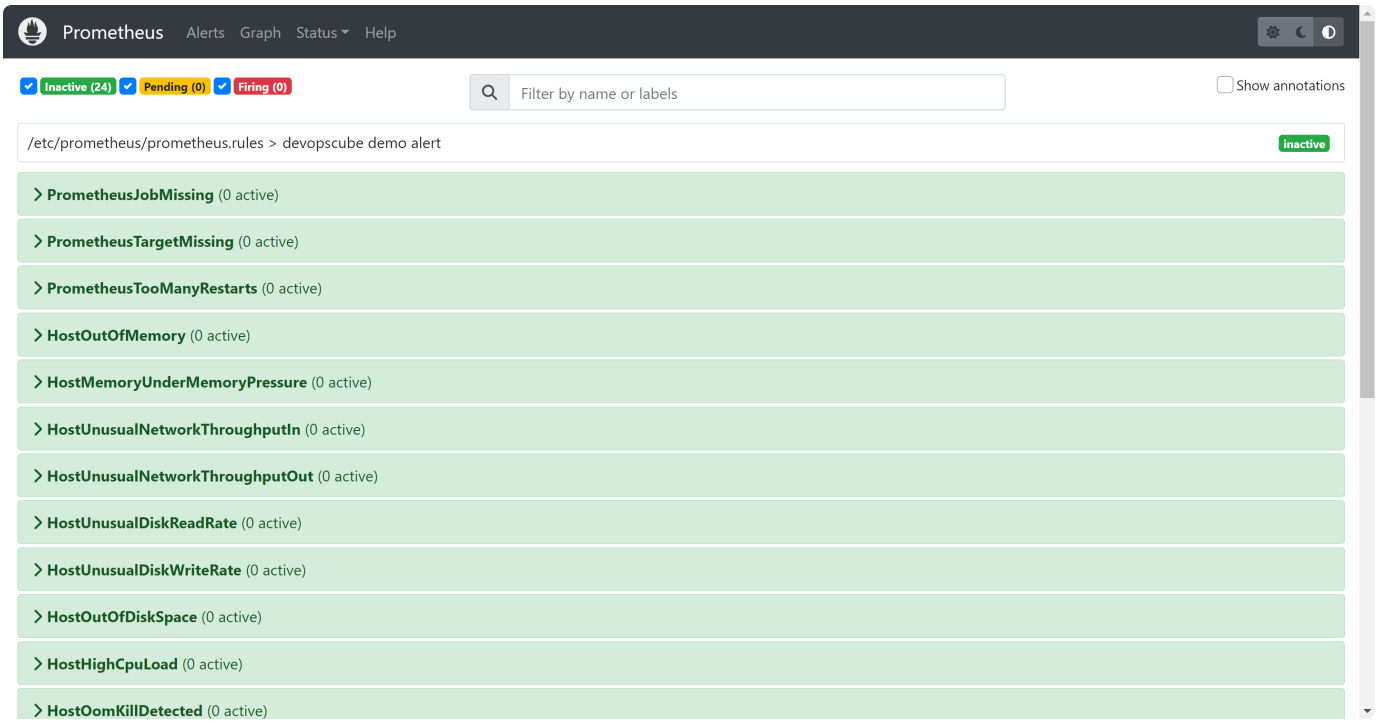
```
        expr: (kube_pod_container_status_restarts_total - kube_pod_container_status_restarts_total offset 10m >= 1) and ignoring (reason) min_over_time(kube_pod_container_status_last_terminated_reason{reason="OOMKilled"}[10m]) == 1
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes Container oom killer (instance {{ $labels.instance }})
          description: "Container {{ $labels.container }} in pod {{ $labels.namespace }}/{{ $labels.pod }} has been OOMKilled {{ $value }} times in the last 10 minutes.\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # Job任务执行failed
      - alert: KubernetesJobFailed
        expr: kube_job_status_failed > 0
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes Job failed (instance {{ $labels.instance }})
          description: "Job {{ $labels.namespace }}/{{ $labels.job_name }} failed to complete\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # 计划任务暂停
      - alert: KubernetesCronjobSuspended
        expr: kube_cronjob_spec_suspend != 0
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes CronJob suspended (instance {{ $labels.instance }})
          description: "CronJob {{ $labels.namespace }}/{{ $labels.cronjob }} is suspended\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # 磁盘PVC未绑定PV
      - alert: KubernetesPersistentvolumeclaimPending
        expr: kube_persistentvolumeclaim_status_phase{phase="Pending"} == 1
        for: 2m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes PersistentVolumeClaim pending (instance {{ $labels.instance }})
          description: "PersistentVolumeClaim {{ $labels.namespace }}/{{ $labels.persistentvolumeclaim }} is pending\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
```

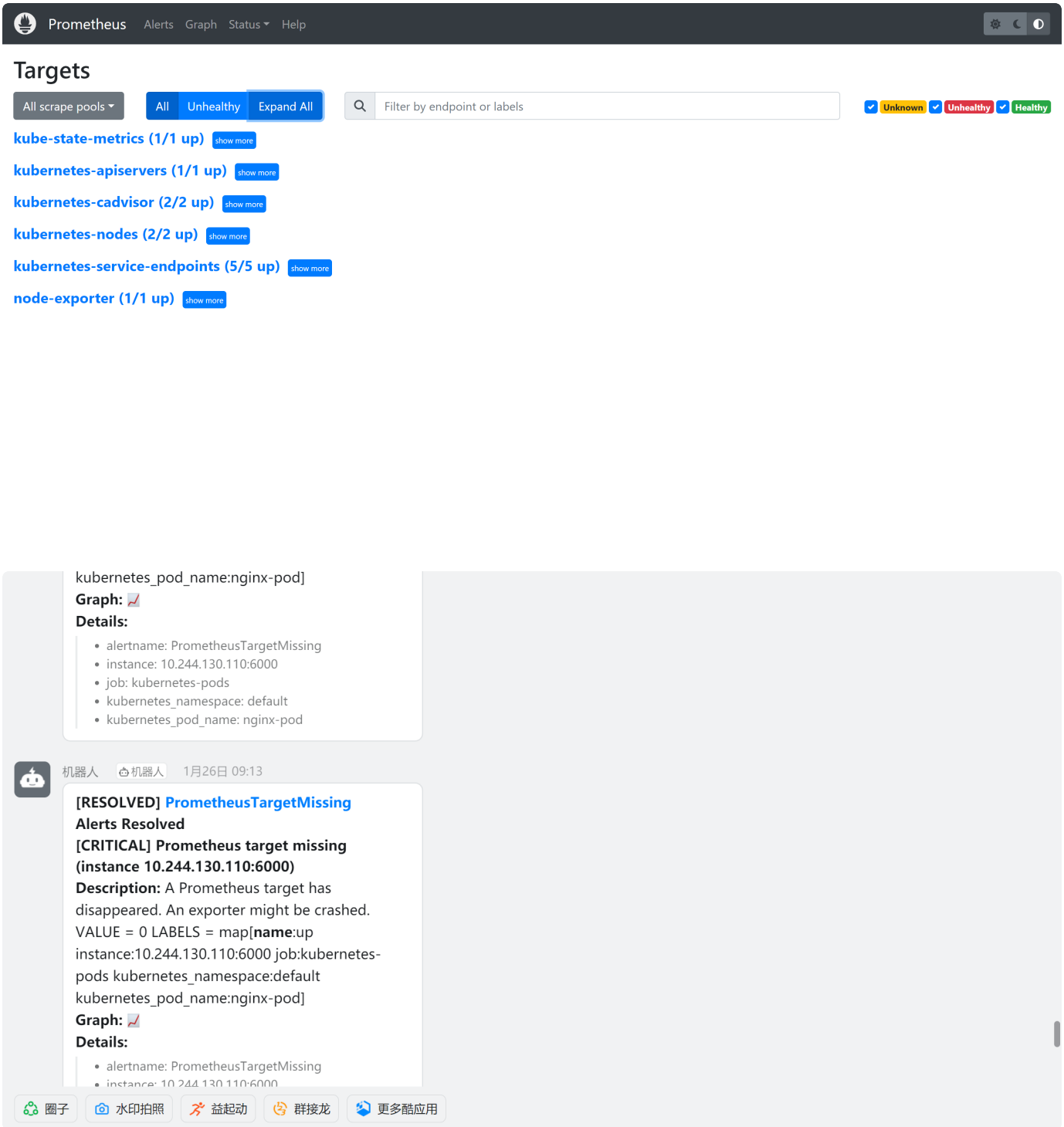
```
    # 挂载卷的存储不足
    - alert: KubernetesVolumeOutOfDiskSpace
      expr: kubelet_volume_stats_available_bytes / kubelet_volume_stats_capacity_bytes * 100 < 10
      for: 2m
      labels:
        severity: warning
      annotations:
        summary: Kubernetes Volume out of disk space (instance {{ $labels.instance }})
        description: "Volume is almost full (< 10% left)\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
    # pv出现异常
    - alert: KubernetesPersistentvolumeError
      expr: kube_persistentvolume_status_phase{phase=~"Failed|Pending", job="kube-state-metrics"} > 0
      for: 0m
      labels:
        severity: critical
      annotations:
        summary: Kubernetes PersistentVolume error (instance {{ $labels.instance }})
        description: "Persistent volume {{ $labels.persistentvolume }} is in bad state\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
    # statefulset控制器出现异常
    - alert: KubernetesStatefulsetDown
      expr: kube_statefulset_replicas != kube_statefulset_status_replicas_ready > 0
      for: 1m
      labels:
        severity: critical
      annotations:
        summary: Kubernetes StatefulSet down (instance {{ $labels.instance }})
        description: "StatefulSet {{ $labels.namespace }}/{{ $labels.statefulset }} went down\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
    # 集群api接口出现异常
    - alert: KubernetesApiServerErrors
      expr: sum(rate(apiserver_request_total{job="kubernetes-apiservers",code=~"^(?:5..)$"}[1m])) / sum(rate(apiserver_request_total{job="apiserver"}[1m])) * 100 > 3
      for: 2m
      labels:
        severity: critical
      annotations:
        summary: Kubernetes API server errors (instance {{ $labels.instance }})
```

```
          description: "Kubernetes API server is experiencing high error rate\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
      # 证书即将过期
      - alert: KubernetesClientCertificateExpiresNextWeek
        expr: apiserver_client_certificate_expiration_seconds_count{job="kubernetes-apiservers"} > 0 and histogram_quantile(0.01, sum by (job, le) (rate(apiserver_client_certificate_expiration_seconds_bucket{job="kubernetes-apiservers"}[5m]))) < 7*24*60*60
        for: 0m
        labels:
          severity: warning
        annotations:
          summary: Kubernetes client certificate expires next week (instance {{ $labels.instance }})
          description: "A client certificate used to authenticate to the apiserver is expiring next week.\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
```

结尾

如果一切顺利的话，可以在Prometheus面板查看所有的的告警项，和集群里的发现目标，有些配置可能需要基于你的集群配置去调整，手册里附带了关于grafana的yaml文件，如果需要可以直接在grafana目录中apply，还附带了所有关于ingress入口的yaml文件，如果你希望通过域名访问Prometheus或者grafana，可以参考ingress目录里的配置进行调整，如果在部署过程中有其他问题，也可以邮箱联系作者 lvtujingji@163.com




Prometheus
Alerts
Graph
Status
Help
Targets
All scrape pools
All
Unhealthy
Expand All
Filter by endpoint or labels
Unknown
Unhealthy
Healthy
kube-state-metrics (1/1 up) show more
kubernetes-apiservers (1/1 up) show more
kubernetes-cadvisor (2/2 up) show more
kubernetes-nodes (2/2 up) show more
kubernetes-service-endpoints (5/5 up) show more
node-exporter (1/1 up) show more
kubernetes_pod_name:nginx-pod]
Graph:
Details:
alertname: PrometheusTargetMissing
instance: 10.244.130.110:6000
job: kubernetes-pods
kubernetes_namespace: default
kubernetes_pod_name: nginx-pod
机器人
机器人
1月26日 09:13
[RESOLVED] PrometheusTargetMissing
Alerts Resolved
[CRITICAL] Prometheus target missing (instance 10.244.130.110:6000)
Description: A Prometheus target has disappeared. An exporter might be crashed. VALUE = 0 LABELS = map[name:up instance:10.244.130.110:6000 job:kubernetes-pods kubernetes_namespace:default kubernetes_pod_name:nginx-pod]
Graph:
Details:
alertname: PrometheusTargetMissing
圈子
水印拍照
益起动
群接龙
更多酷应用