

ЗАДАНИЕ № 7

1) Разверните в кластере сервер систему мониторинга Prometheus. Создайте в кластере ConfigMap со следующим содержимым:

```
prometheus.yml: |
global:
  scrape_interval: 30s

scrape_configs:
- job_name: 'prometheus'
static_configs:
- targets: ['localhost:9090']

- job_name: 'kubernetes-nodes'
  kubernetes_sd_configs:
  - role: node
  relabel_configs:
  - source_labels: [__address__]
    regex: (.+):(.+)
    target_label: __address__
    replacement: ${1}:9101
```

Создайте объекты для авторизации Prometheus сервера в Kubernetes-API:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: prometheus

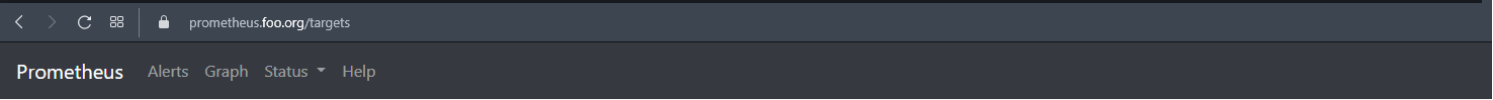
namespace: default
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: prometheus
rules:
- apiGroups: [""]
resources:
- nodes
```

```
verbs: ["get", "list", "watch"]
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: prometheus
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: prometheus
subjects:
- kind: ServiceAccount
  name: prometheus
  namespace: default
```

Создайте StatefulSet для Prometheus сервера из образа prom/prometheus:v2.19.2 с одной репликой. В нем должен быть описан порт 9090 TCP volumeClaimTemplate - ReadWriteOnce, 5Gi, подключенный по пути /prometheus Подключение конфигмапа с настройками выше по пути /etc/prometheus. Так же в этом стейтфулсете нужно объявить initContainer для изменения прав на вольюм data на 777. См пример из лекции 4: practice/4.resources-and-persistence/persistence/deployment.yaml. Не забудьте указать обязательное поле serviceName. Так же укажите поле serviceAccount: prometheus на одном уровне с containers, initContainers, volumes См пример с rabbitmq из материалов лекции. Создайте service и ingress для этого стейтфулсета, так чтобы запросы с любым доменом на белый IP вашего сервиса nginx-ingress-controller (тот что в нэймспэйсе ingress-nginx с типом LoadBalancer) шли на приложение. Проверьте что при обращении из браузера на белый IP вы видите открывшееся приложение Prometheus. В этом же неймспэйсе создайте DaemonSet node-exporter как в примере к лекции: practice/7.advanced-abstractions/daemonset.yaml. Откройте в браузере интерфейс Prometheus. Попробуйте открыть Status -> Targets. Тут вы должны увидеть все ноды своего кластера, которые Prometheus смог определить и собирает с ним метрики. Так же можете попробовать на вкладке Graph выполнить запрос node_load1 - это минутный Load Average для каждой из нод в кластере.

РЕШЕНИЕ:

1)



Targets

All Unhealthy

kubernetes-nodes (3/3 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://10.0.0.4:9101/metrics	UP	instance="kubernetes-cluster-7566-master-0" job="kubernetes-nodes"	2.902s ago	14.04ms	
http://10.0.0.9:9101/metrics	UP	instance="kubernetes-cluster-7566-worker-group-0" job="kubernetes-nodes"	15s ago	11.68ms	
http://10.0.0.12:9101/metrics	UP	instance="kubernetes-cluster-7566-worker-group-1" job="kubernetes-nodes"	5.13s ago	11.3ms	

prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	5.352s ago	3.466ms	

