Домашнее задание для к уроку 7 Продвинутые абстракции Kubernetes!

! Задание нужно выполнять в нэймспэйсе default 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
```

• Coздайте StatefulSet для Prometheus сервера из образа prom/prometheus:v2.19.2 с одной репликой

В нем должнен 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.

minikube status

```
т tolstikov@my3vm:~-Терминал — + × Файл Правка Вид Терминал Вкладки Справка tolstikov@my3vm:~$ minikube status

| Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 2.698707275s
| Restarting the docker service may improve performance. minikube type: Control Plane host: Running kubelet: Running apiserver: Running apiserver: Running kubeconfig: Configured tolstikov@my3vm:~$ ■
```

kubectl cluster-info

```
то tolstikov@my3vm:~-Терминал — + х Файл Правка Вид Терминал Вкладки Справка tolstikov@my3vm:~$ kubectl cluster-info Киbernetes control plane is running at https://192.168.49.2:8443 CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'. tolstikov@my3vm:~$ ■
```

vi configmap.yaml

```
tolstikov@my3vm: ~ - Терминал
 Файл Правка
               Вид Терминал Вкладки
   2 apiVersion: v1
   3 kind: ConfigMap
   4 metadata:
   5 name: prometheus-config
      namespace: default
   7 data:
      prometheus.yml: |
   9
        global:
  10
           scrape interval: 30s
         scrape_configs:
  11
           - job_name: 'prometheus'
  12
  13
             static_configs:
             - targets: ['localhost:9090']
  14
           job_name: 'kubernetes-nodes'
kubernetes_sd_configs:
  15
  16
  17
              - role: node
             relabel configs:
  18
             source_labels: [_address_]
  19
               regex: (.+):(.+)
  20
               target label:
  21
                                address
               replacement: \$\{1\}:9101
  22
:wq
```

kubectl apply -f configmap.yaml

```
▼ tolstikov@my3vm:~-Терминал — + ×
Файл Правка Вид Терминал Вкладки Справка

Press ENTER or type command to continue
tolstikov@my3vm:~$ kubectl apply -f configmap.yaml
configmap/prometheus-config created
tolstikov@my3vm:~$
```

vi serviceaccount.yaml

```
▼ tolstikov@my3vm:~-Терминал — + ×
Файл Правка Вид Терминал Вкладки Справка

1 ---
2 apiVersion: v1
3 kind: ServiceAccount
4 metadata:
5 name: prometheus
6 namespace: default
~
:wq
```

kubectl apply -f serviceaccount.yaml

```
▼ tolstikov@my3vm:~-Терминал — + ×

Файл Правка Вид Терминал Вкладки Справка

configmap/prometheus-config created

tolstikov@my3vm:~$ vi serviceaccount.yaml

Error detected while processing /home/tolstikov/.vimrc:

line 3:

E487: Argument must be positive: tabstop=0

Press ENTER or type command to continue

tolstikov@my3vm:~$ kubectl apply -f serviceaccount.yaml

serviceaccount/prometheus created

tolstikov@my3vm:~$
```

vi clusterrole.yaml

```
tolstikov@my3vm:~-Терминал — + х
Файл Правка Вид Терминал Вкладки Справка

1 ---
2 apiVersion: rbac.authorization.k8s.io/v1
3 kind: ClusterRole
4 metadata:
5 name: prometheus
6 rules:
7 - apiGroups: [""]
8 resources:
9 - nodes
10 verbs: ["get", "list", "watch"]
:wq
```

kubectl apply -f clusterrole.yaml

```
то tolstikov@my3vm:~-Терминал — + × Файл Правка Вид Терминал Вкладки Справка

Error detected while processing /home/tolstikov/.vimrc:
line 3:

E487: Argument must be positive: tabstop=0

Press ENTER or type command to continue
tolstikov@my3vm:~$ kubectl apply -f clusterrole.yaml
clusterrole.rbac.authorization.k8s.io/prometheus created
tolstikov@my3vm:~$
```

vi clusterrolebinding.yaml

```
tolstikov@my3vm: ~ - Терминал
Файл Правка Вид Терминал Вкладки Справка
  2 apiVersion: rbac.authorization.k8s.io/vl
  3 kind: ClusterRoleBinding
  4 metadata:
  5 name: prometheus
  6 roleRef:
      apiGroup: rbac.authorization.k8s.io
  8 kind: ClusterRole
     name: prometheus
 10 subjects:
 11 - kind: ServiceAccount
  12
      name: prometheus
 13
     namespace: default
```

kubectl apply -f clusterrolebinding.yaml

```
тоlstikov@my3vm:~-Терминал — + ×
Файл Правка Вид Терминал Вкладки Справка
tolstikov@my3vm:~$ kubectl apply -f clusterrolebinding.yaml
clusterrolebinding.rbac.authorization.k8s.io/prometheus created
tolstikov@my3vm:~$
```

vi statefulset.yaml $- + \times$ tolstikov@my3vm: ~ - Терминал Файл Правка Вид Терминал Вкладки Справка 1 ---2 apiVersion: apps/vl 3 kind: StatefulSet 4 metadata: 5 name: prometheus 6 namespace: default 7 spec: 8 serviceName: prometheus 9 replicas: 1 10 selector: 11 matchLabels: 12 app: prometheus 13 template: 14 metadata: 15

```
▼ tolstikov@my3vm:~-Терминал — + ×
Файл Правка Вид Терминал Вкладки Справка

Press ENTER or type command to continue
tolstikov@my3vm:~$ kubectl apply -f statefulset.yaml
statefulset.apps/prometheus configured
tolstikov@my3vm:~$
```

vi service.yaml

```
- + ×
                    tolstikov@my3vm: ~ - Терминал
Файл Правка Вид Терминал Вкладки Справка
   1 ---
   2 kind: Service
  3 apiVersion: vl
  4 metadata:
  5 name: prometheus
  6 labels:
        app: prometheus
  8 spec:
  9 ports:
      - protocol: TCP
  10
  11
          port: 80
  12 targetPort: 9090
  13 selector:
  14 app: prometheus
:wq
```

kubectl apply -f service.yaml

```
▼ tolstikov@my3vm:~-Терминал — + ×
Файл Правка Вид Терминал Вкладки Справка

E487: Argument must be positive: tabstop=0

Press ENTER or type command to continue
tolstikov@my3vm:~$ kubectl apply -f service.yaml
service/prometheus created
tolstikov@my3vm:~$
```

vi daemonset.yaml

```
tolstikov@my3vm: ~ - Терминал
Файл Правка Вид Терминал Вкладки Справка
    2 apiVersion: kubernetes.io/os
3 kind: DaemonSet
4 metadata:
5 labels:
             app: node-exporter
name: node-exporter
     8 spec:
9 upd
              updateStrategy:
                   rollingUpdate:
maxUnavailable: 1
   11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
             type: RollingUpdate selector: matchLabels:
              app: node-exporter template:
                   metadata:
labels:
                  app: node-exporter
spec:
                         containers:
                         - args:
         - args:
- -web.listen-address=0.0.0.0:9101
- --path.procfs=/host/proc
- -path.sysfs=/host/sys
- --collector.filesystem.ignored-mount-points=^/(dev|proc|sys|var/lib/docker/.+)($|/)
- --collector.filesystem.ignored-fs-types=^(autofs|binfmt_misc|cgroup|configfs|debugfs|devpts|devtmpfs|fusectl|hugetlbfs|mqueue|overlay|
proc|procfs|pstore|rpc_pipefs|securityfs|sysfs|tracefs)$
image: quay.io/prometheus/node-exporter:v0.16.0
imagePullPolicy: IfNotPresent
   28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
57
58
                            name: node-exporter
volumeMounts:
- mountPath: /host/proc
name: proc
- mountPath: /host/sys
                             name: sys
- mountPath: /host/root
                         name: root
readOnly: true
hostNetwork: true
                        hostPID: true
tolerations:
- effect: NoSchedule
operator: Exists
nodeSelector:
                        beta.kubernetes.io/os: linux
volumes:
                       - hostPath:
path: /proc
type: ""
name: proc
- hostPath:
                       - hostPath:
path: /sys
type: ""
name: sys
- hostPath:
path: /
type: ""
name: root
```

kubectl create -f daemonset.yaml

```
tolstikov@my3vm:~-Терминал — + ×
Файл Правка Вид Терминал Вкладки Справка
tolstikov@my3vm:~$ kubectl create -f daemonset.yaml
Warning: spec.template.spec.nodeSelector[beta.kubernetes.io/os]: deprecated since v1.14; use "kubernetes.io/os" instead
daemonset.apps/node-exporter created
tolstikov@my3vm:~$

■
```

vi ingress.yaml

```
+
                       tolstikov@my3vm: ~ - Терминал
                                                                                ×
Файл
       Правка
                Вид
                      Терминал Вкладки
                                           Справка
   1 ---
    apiVersion: networking.k8s.io/v1
   3 kind: Ingress
   4 metadata:
       name: prometheus
  6 spec:
       rules:
       - http:
  8
  9
           paths:
           - path: /prometheus
 10
 11
             pathType: Prefix
 12
             backend:
 13
               service:
 14
                 name: prometheus
 15
                 port:
 16
                   number: 80
                                                                            All
                                                              1,1
```

kubectl apply -f ingress.yaml

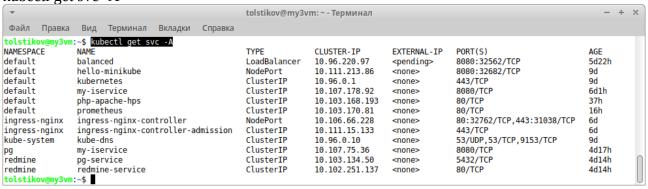
```
tolstikov@my3vm:~-Терминал — + ×
Файл Правка Вид Терминал Вкладки Справка

E487: Argument must be positive: tabstop=0
Press ENTER or type command to continue
tolstikov@my3vm:~$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/prometheus created
tolstikov@my3vm:~$ |
```

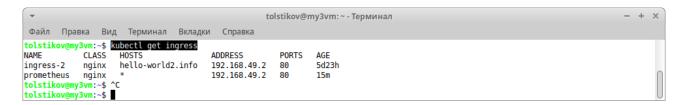
kubectl get pod

```
tolstikov@my3vm: ~ - Терминал
                                                                             4-
                                                                                30
 Файл
        Правка
                 Вид
                       Терминал Вкладки
                                             Справка
E487: Argument must be positive: tabstop=0
Press ENTER or type command to continue
tolstikov@my3vm:~$ kubectl get pod
NAME
                               STATUS
                                         RESTARTS
                                                     AGE
                      READY
node-exporter-pb8lk
                               Running
                                                     16h
                      1/1
                                         0
                                                     17h
prometheus-0
                      0/1
                               Pending
                                         0
tolstikov@my3vm:~$
```

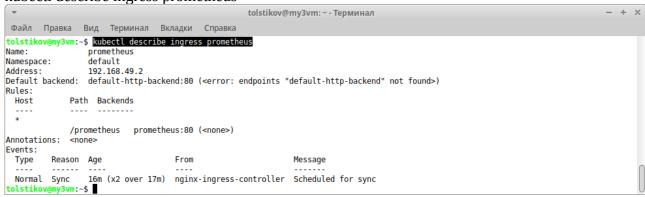
kubectl get svc -A



kubectl get ingress



kubectl describe ingress prometheus



kubectl get jobs



kubectl describe pod prometheus-0

```
igor@ubuntu-server: ~ - Терминал
 Файл Правка Вид Терминал Вкладки Справка
igor@ubuntu-server:~$ kubectl describe pod prometheus-0
                 default
Namespace:
Priority:
                  <none>
                 app=prometheus
controller-revision-hash=prometheus-77d44fd74b
statefulset.kubernetes.io/pod-name=prometheus-0
Labels:
Annotations:
Status:
IP:
Controlled By: StatefulSet/prometheus
Init Containers:
mount-permissions-fix:
    Image:
                 busybox
    Port: <none>
    Command:
      sh
      chmod 777 /prometheus
    Environment: <none>
      /prometheus from data (rw)
       /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-k97f8 (ro)
Containers:
  prometheus-k8s:
                   prom/prometheus:v2.19.2
9090/TCP
    Image:
    Port:
    Host Port: 0/TCP
Environment: <none>
    Mounts:
       /etc/prometheus from config (rw)
       /prometheus from data (rw)
       /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-k97f8 (ro)
Conditions:
  Type
PodScheduled
                  Status
Volumes:
  data:
                 PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    Type:
ClaimName:
    ReadOnly:
  confia:
                ConfigMap (a volume populated by a ConfigMap)
    Type:
    Name: prometheus-config
Optional: false
  kube-api-access-k97f8:
    Type:
TokenExpirationSeconds:
                                Projected (a volume that contains injected data from multiple sources) 3607
    ConfigMapName:
                                kube-root-ca.crt
    ConfigMapOptional:
DownwardAPI:
                                <nil>
                                true
Oos Class:
                                BestEffort
                                node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
Tolerations:
                                node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
                                                                             Message
  Warning FailedScheduling 3m25s (x66 over 68m) default-scheduler 0/1 nodes are available: 1 pod has unbound immediate PersistentVolumeClaims.
```

kubectl delete pod node-exporter-lxv2v

```
igor@ubuntu-server:~-Терминал — + × Файл Правка Вид Терминал Вкладки Справка

igor@ubuntu-server:-$ kubectl get pods

NAME READY STATUS RESTARTS AGE

node-exporter-lxv2v 1/1 Running 0 62m

prometheus-0 0/1 Pending 0 72m

igor@ubuntu-server:-$ kubectl delete pod node-exporter-lxv2v

pod "node-exporter-lxv2v" deleted

igor@ubuntu-server:-$ ■
```

kubectl get replicaset

```
т igor@ubuntu-server:~-Терминал — + ×

Файл Правка Вид Терминал Вкладки Справка

igor@ubuntu-server:~$ kubectl get pods

NAME READY STATUS RESTARTS AGE
node-exporter-ps8xs 1/1 Running 0 75s
prometheus-0 0/1 Pending 0 74m
igor@ubuntu-server:~$ kubectl get replicaset
NO resources found in default namespace.
igor@ubuntu-server:~$
```



kubectl get daemontset



kubectl top pod

