

## 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
```

- 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 конфигама 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 неймспэйсе 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

```
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
kubeconfig: Configured
tolstikov@my3vm:~$
```

kubectl cluster-info

```
tolstikov@my3vm: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
tolstikov@my3vm:~$ kubectl cluster-info
Kubernetes 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  Справка

1 ---
2 apiVersion: v1
3 kind: ConfigMap
4 metadata:
5   name: prometheus-config
6   namespace: default
7 data:
8   prometheus.yml: |
9     global:
10      scrape_interval: 30s
11      scrape_configs:
12        - job_name: 'prometheus'
13          static_configs:
14            - targets: ['localhost:9090']
15        - job_name: 'kubernetes-nodes'
16          kubernetes_sd_configs:
17            - role: node
18          relabel_configs:
19            - source_labels: [__address__]
20              regex: (.+):(.+)
21              target_label: __address__
22              replacement: ${1}:9101
~
~
: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
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"]
:wd
```

kubectl apply -f clusterrole.yaml

```
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
1 ---
2 apiVersion: rbac.authorization.k8s.io/v1
3 kind: ClusterRoleBinding
4 metadata:
5   name: prometheus
6 roleRef:
7   apiGroup: rbac.authorization.k8s.io
8   kind: ClusterRole
9   name: prometheus
10 subjects:
11 - kind: ServiceAccount
12   name: prometheus
13   namespace: default
~
:wd
```

kubectl apply -f clusterrolebinding.yaml

```
tolstikov@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

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

1  ---
2  apiVersion: apps/v1
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       labels:
16         app: prometheus
17     spec:
18       serviceAccount: prometheus
19       terminationGracePeriodSeconds: 10
20       initContainers:
21         - image: busybox
22           name: mount-permissions-fix
23           command: [ "sh", "-c", "chmod 777 /prometheus" ]
24           volumeMounts:
25             - name: data
26               mountPath: /prometheus
27       containers:
28         - name: prometheus-k8s
29           image: prom/prometheus:v2.19.2
30           ports:
31             - protocol: TCP
32               containerPort: 9090
33           imagePullPolicy: IfNotPresent
34           volumeMounts:
35             - name: config
36               mountPath: /etc/prometheus
37             - name: data
38               mountPath: /prometheus
39       volumes:
40         - name: config
41           configMap:
42             name: prometheus-config
43       volumeClaimTemplates:
44         - metadata:
45             name: data
46           spec:
47             accessModes: [ "ReadWriteOnce" ]
48             resources:
49               requests:
50                 storage: 5Gi
51             storageClassName: csi-ceph-hdd-dp1
~
~
:wq
```

kubectl apply -f statefulset.yaml

```
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: v1
4 metadata:
5   name: prometheus
6   labels:
7     app: prometheus
8 spec:
9   ports:
10    - protocol: TCP
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

1 ---
2 apiVersion: kubernetes.io/os
3 kind: DaemonSet
4 metadata:
5   labels:
6     app: node-exporter
7   name: node-exporter
8 spec:
9   updateStrategy:
10     rollingUpdate:
11       maxUnavailable: 1
12     type: RollingUpdate
13   selector:
14     matchLabels:
15       app: node-exporter
16   template:
17     metadata:
18       labels:
19         app: node-exporter
20     spec:
21       containers:
22         - args:
23             - --web.listen-address=0.0.0.0:9101
24             - --path.procfs=/host/proc
25             - --path.sysfs=/host/sys
26             - --collector.filesystem.ignored-mount-points=^/(dev|proc|sys|var/lib/docker/.+)($/|)
27             - --collector.filesystem.ignored-fs-types=^(autofs|binfmt_misc|cgroup|configfs|debugfs|devpts|devtmpfs|fusectl|hugetlbfs|mqueue|overlay|
proc|procfs|pstore|rpc_pipefs|securityfs|sysfs|tracefs)$
28           image: quay.io/prometheus/node-exporter:v0.16.0
29           imagePullPolicy: IfNotPresent
30           name: node-exporter
31           volumeMounts:
32             - mountPath: /host/proc
33               name: proc
34             - mountPath: /host/sys
35               name: sys
36             - mountPath: /host/root
37               name: root
38             readOnly: true
39           hostNetwork: true
40           hostPID: true
41           tolerations:
42             - effect: NoSchedule
43               operator: Exists
44           nodeSelector:
45             beta.kubernetes.io/os: linux
46           volumes:
47             - hostPath:
48                 path: /proc
49                 type: ""
50               name: proc
51             - hostPath:
52                 path: /sys
53                 type: ""
54               name: sys
55             - hostPath:
56                 path: /
57                 type: ""
58               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  --
2  apiVersion: networking.k8s.io/v1
3  kind: Ingress
4  metadata:
5    name: prometheus
6  spec:
7    rules:
8      - http:
9        paths:
10         - path: /prometheus
11           pathType: Prefix
12           backend:
13             service:
14               name: prometheus
15               port:
16                 number: 80

1,1  All
```

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
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                READY   STATUS    RESTARTS   AGE
node-exporter-pb8lk  1/1     Running   0           16h
prometheus-0         0/1     Pending   0           17h
tolstikov@my3vm:~$
```

kubectl get svc -A

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

tolstikov@my3vm:~$ kubectl get svc -A
NAMESPACE   NAME                TYPE          CLUSTER-IP      EXTERNAL-IP  PORT(S)                AGE
default     balanced            LoadBalancer  10.96.220.97    <pending>    8080:32562/TCP         5d22h
default     hello-minikube      NodePort      10.111.213.86   <none>       8080:32682/TCP         9d
default     kubernetes          ClusterIP     10.96.0.1       <none>       443/TCP                9d
default     my-iservice         ClusterIP     10.107.178.92   <none>       8080/TCP                6d1h
default     php-apache-hps      ClusterIP     10.103.168.193  <none>       80/TCP                 37h
default     prometheus          ClusterIP     10.103.170.81   <none>       80/TCP                 16h
ingress-nginx ingress-nginx-controller NodePort      10.106.66.228   <none>       80:32762/TCP,443:31030/TCP 6d
ingress-nginx ingress-nginx-controller-admission ClusterIP      10.111.15.133   <none>       443/TCP                 6d
kube-system kube-dns            ClusterIP     10.96.0.10      <none>       53/UDP,53/TCP,9153/TCP  9d
pg          my-iservice         ClusterIP     10.107.75.36    <none>       8080/TCP                4d17h
redmine     pg-service          ClusterIP     10.103.134.50   <none>       5432/TCP                4d14h
redmine     redmine-service     ClusterIP     10.102.251.137  <none>       80/TCP                 4d14h
tolstikov@my3vm:~$
```

kubectl get ingress



```
tolstikov@my3vm: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
tolstikov@my3vm:~$ kubectl get ingress
NAME      CLASS    HOSTS                ADDRESS      PORTS    AGE
ingress-2  nginx    hello-world2.info    192.168.49.2  80       5d23h
prometheus nginx    *                    192.168.49.2  80       15m
tolstikov@my3vm:~$ ^C
tolstikov@my3vm:~$
```

kubectl describe ingress prometheus

```
tolstikov@my3vm: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
tolstikov@my3vm:~$ kubectl describe ingress prometheus
Name:      prometheus
Namespace: default
Address:    192.168.49.2
Default backend: default-http-backend:80 (<error: endpoints "default-http-backend" not found>)
Rules:
  Host      Path  Backends
  ----      -
  *          /prometheus  prometheus:80 (<none>)
Annotations: <none>
Events:
  Type    Reason    Age          From              Message
  ----    -
  Normal  Sync      16m (x2 over 17m)  nginx-ingress-controller  Scheduled for sync
tolstikov@my3vm:~$
```

kubectl get jobs

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl get jobs
No resources found in default namespace.
igor@ubuntu-server:~$
```

kubectl describe pod prometheus-0

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

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

kubectl delete pod node-exporter-lxv2v

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

igor@ubuntu-server:~$ kubectl get pods
NAME                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

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

igor@ubuntu-server:~$ kubectl get pods
NAME                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 deployments

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
No resources found in default namespace.
igor@ubuntu-server:~$ kubectl get deployments
No resources found in default namespace.
igor@ubuntu-server:~$
```

kubectl get daemonset

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl get daemonset
error: the server doesn't have a resource type "daemonset"
igor@ubuntu-server:~$
```

kubectl top pod

```
igor@ubuntu-server: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@ubuntu-server:~$ kubectl top pod
error: Metrics API not available
igor@ubuntu-server:~$
```