

microk8s

Оркестрация

- [Dashboard ничего не показывает](#)
- [Получение конфигурации для Lens](#)
- [Lens - Metrics not available at the moment](#)
- [Unable to connect to the server: x509: certificate has expired](#)
- [Удаление завершённых работ](#)
- [Прямой вызов kubectl](#)
- [Удаление подов в статусе Error](#)
- [Сертификаты](#)
 - [Проверка сертификатов](#)
 - [Обновление сертификатов](#)
- [Погасить все поды](#)
- [Кастомные порты Ingress](#)

Dashboard ??????? ?? ??????????????

После установки и включения dashboard (ссылка на статью <https://microk8s.io/docs/addon-dashboard>) ничего не отображается по причине отсутствия прав.

Не забываем:

```
microk8s kubectl port-forward --address 0.0.0.0 -n kube-system service/kubernetes-dashboard 10443:443
```

Чтобы все заработало, необходимо сделать следующее:

Создать файл `defaultrolebinding.yaml` со следующим содержимым:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: default-role-binding
subjects:
- kind: ServiceAccount
  name: default
  namespace: kube-system
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
```

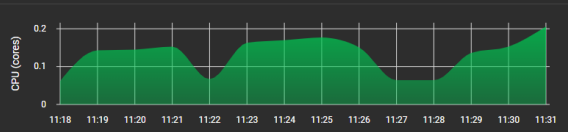
Выполнить следующую команду:

```
microk8s kubectl apply -f defaultrolebinding.yaml
```

В результате мы увидим информацию о своем узле (кластере):

Workloads

Config and Storage



The following table represents the data shown in the donut charts:

Object Type	Count
Pods	10
Deployments	5
Replica Sets	4
Stateful Sets	3
Jobs	2

????????? ?????????????? ???

Lens

Для подключения к кластеру с помощью Lens необходимо получить его конфигурацию. Для этого необходимо ввести в консоли команду:

```
microk8s config
```

Результатом выполнения будет информация, которую необходимо скопировать в Lens при добавлении кластера:

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURCRENDQWV5Z0F3SUJBZ0lKQVBQMghoaFduTTR0TUEwR0NTcUdTZW
IzRFFQkN3VUFNQmN4RlRBVEJnTlYkQkFNTURERXdMakUxTWk0eE9ETXVNVFElRncweE9URXdNVFl3T1RRNE5UaGFGdzAw
TnpBek1ETXdPVFE0TlRoYQpNQmN4RlRBVEJnTlZCQU1NRERFd0xqRTFNaTR4T0RNdU1UQ0NBUE0l3RFFZSkvWklodmNOQV
FFQkJRQRnZ0VQCKFEQ0NBUEW9DZ2dFQkFQbWpiZ3F6NWxTL3FlTldlcDduVE9NUS94NXp6MUI5bGR3dTJkZkVxbEtIRXVW
T3RZYU0KeVJMb0xtYmZBbUxTd2ZDSXIycWhyZ1UxTlFTd096WUVCeXpNZ01WaWtzejZlVjJ3Um10b3RxbXpXLzZHQTl1SQ
pmM1MvTURERHh2ZmQva1hrb2lpeTE5WCtPVENDTHFUTmJZRG45RkdRTjJWSVZzVWV5Z255LzJjbkpDb0xLTGtBCm04NHVT
UXBLYUJ1T3hzeFZBU3J6cURINK1uczN5OpyWXRqSG9BaDBTcWxEdHhKb1pGOGJpTGx2Q2lRVkMyNE4KQ2IyMkpRMHVyQW
FRUlbIRmRsaGtKU1RBL2Y0VFp1VDN2N2tpaU9QZWpGT252M0lXYld3UXZRYU5ad0JmM0NQwozK0dIRXNSU1NPYkQ5Z2FG
ODhVTXNScEM1TFBaVm12eFFVc0NBd0VBQWF0VE1GRXdIUUVlEVlIiwT0JCWUUVGUElKComp6ZWhaYW1nS0xPSkQyVitiU3VMaH
FjU01C0EdBMVvkSXdRWU1CYUFGUElKanplaFphbWdLTE9KRdJWk2JTdUwKaHFjU01B0EdBMVvkRXdFQi93UUZNQU1CQWY4
d0RRWUpLb1pJaHJzTkFRRUxCUUFEEZ2dFQkFI0VlPaGRMT3RXKwpxbkhnajRYMjcwSWRFNDNlc3JRRENEUnY4UGpRVmVwYy
tNUEdrMU10cFNsckgvTkNnc2puMzZ2RmpFeUJzcXFJCKh4QmhPdi9wR3NIUhdUa2p3bXNSYkdwWkliQTdRcGNKSTZlaEM4
aFUwSmI2b3ZwSU9zZnR6SFF6SVA4NWRh0EsKWGptbVY2UzNLQnZtZlpKMU1WZzFuUFBGYnpBdVhoaXI0RmFvZ3AzL2lwNk
VFdWtYNUYydnQvVDDSaC9XODFDNQphUFhpTDdSMTUxQm9aU3U2Y2ZUdGx2Y092cW9DSDFvaXlvWlJIV2dSeElPVTU2dXVu
RXlUdVNoMjJnbGVSSlAyCk8xWk9rNEVBS3VNMEsra05SeEVvcXNmeFlilYlZicENvbGVJRXLlUUhNzJJU0RrdlVWRmhRMk
xGZTJ1bmtvRDgKY2JZbVAXVU1tZ009Ci0tLS0tRU5EIEFUF1RJRklDQVRFLS0tLS0K

    server: https://192.168.0.2:16443
    name: microk8s-cluster
contexts:
- context:
    cluster: microk8s-cluster
    user: admin
```

```
  name: mikro8s
current-context: mikro8s
kind: Config
preferences: {}
users:
- name: admin
  user:
    username: admin
    password: S05BcnhqaWpCNzlRaWtDZEIzNHEzVU5qKzZCaERUbWxjQ2d6VTFmcjkrQT0K
```

Lens - Metrics not available at the moment

For any of you who have the same issue, you can use my ClusterRole.yml here which is based on the PR:

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: lens-read-metrics
rules:
  - apiGroups:
      - ""
    resources:
      - services/proxy
    verbs:
      - get
      - list
      - watch
```

You can bind this ClusterRole with your newly created service accounts with:

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: lens-read-metrics
  namespace: lens-metrics # this is the namespace where Lens creates its metrics services by default
subjects:
  - kind: ServiceAccount
    name: <put your account name here>
    namespace: <your service account namespace>
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: lens-read-metrics
```

Unable to connect to the server: x509: certificate has expired

Не стартуют сервисы. При попытке что либо сделать получаем проблему с сертификатом, например:

```
microk8s kubectl get pod
```

Unable to connect to the server: x509: certificate has expired or is not yet valid: current time 2022-05-18T12:09:52+03:00 is after 2022-04-28T07:26:01Z

Как лечить:

```
microk8s.refresh-certs --cert server.crt
```

?????? ???? ???? ???? ?

```
microk8s.kubectl delete jobs --field-selector status.successful=1
```


??????? ?????? kubectl

При установке через snap microk8s kubectl "в лоб" недоступен. Чтобы исправить данную проблему следует выполнить:

```
snap alias microk8s.kubectl kubectl
```

теперь можно использовать kubectl напрямую.

????????? ?????? ? ??????????

Error

```
kubectl get pods --field-selector 'status.phase=Failed' -o name | xargs kubectl delete
```

??????????

Сертификаты

???????? ???? ???? ???? ???? ???? ???? ???? ?

```
microk8s.refresh-certs -c
```

Сертификаты

????????? ??????????

После [проверки](#) сертификатов возможно потребуется их обновить. Возможно несколько вариантов, например:

The CA certificate will expire in 3282 days.

The server certificate will expire in -3 days.

The front proxy client certificate will expire in -3 days.

Тогда нужно обновить серверный и фронт сертификаты командами:

```
microk8s.refresh-certs -e server.crt  
microk8s.refresh-certs -e front-proxy-client.crt
```

Но лучше зафиксировать это в crone:

```
5 1 1 * * /snap/bin/microk8s.refresh-certs -e server.crt  
5 1 2 * * /snap/bin/microk8s.refresh-certs -e front-proxy-client.crt
```

???????? ???? ?????

```
kubectl scale deploy -n <namespace> --replicas=0 --all
```

????????? ????? Ingress

```
kubectrl patch ds/nginx-ingress-microk8s-controller -n ingress --type='json' -p='[{"op":  
"replace", "path": "/spec/template/spec/containers/0/ports",  
"value":[{"containerPort":80,"hostPort":8080,"name":"http","protocol":"TCP"}, {"containerPort":  
443,"hostPort":8443,"name":"https","protocol":"TCP"}, {"containerPort":10254,"hostPort":10254,"  
name":"health","protocol":"TCP"}]}'
```

```
kubectrl rollout restart ds/nginx-ingress-microk8s-controller -n ingress
```