

microk8s

Данный способ установки является устаревшим

- [Dashboard ничего не показывает](#)
- [unable to recognize "": no matches for kind "Ingress" in version "extensions/v1beta1"](#)
- [Получение конфигурации для Lens](#)
- [Lens - Metrics not available at the moment](#)
- [Unable to connect to the server: x509: certificate has expired](#)
- [Удаление завершенных работ](#)
- [Прямой вызов kubectl](#)
- [Удаление подов в статусе Error](#)
- [Сертификаты](#)
 - [Проверка сертификатов](#)
 - [Обновление сертификатов](#)
- [Погасить все поды](#)
- [Кастомные порты Ingress](#)
- [Установка с нуля](#)
 - [1. Базовая](#)
 - [2. Metric Server](#)
 - [3. Патчим ingress](#)
- [Очистка от подов в статусе Terminating](#)
- [Установка в LXC](#)

Dashboard ничего не показывает

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:

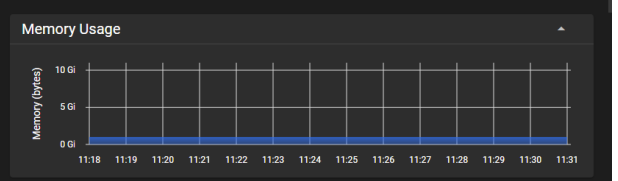
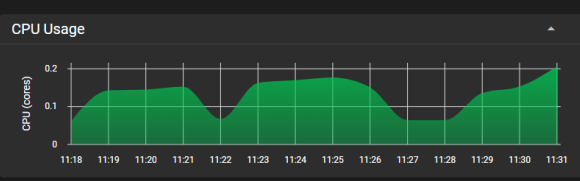
```
microk8s kubectl apply -f defaultrolebinding.yaml
```

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

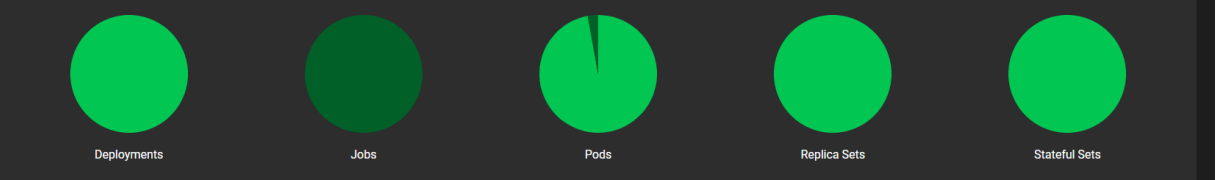
Overview

- Cluster
- Cluster Roles
- Namespaces
- Nodes
- Persistent Volumes
- Storage Classes
- Namespace
- default
- Overview
- Workloads
 - Cron Jobs
 - Daemon Sets
 - Deployments
 - Jobs
 - Pods
 - Replica Sets
 - Replication Controllers
 - Stateful Sets
- Discovery and Load Balancing
 - Ingresses
 - Services
- Config and Storage

Workloads



Workload Status



Deployments

| Name | Namespace | Labels | Pods | Created | Images |
|------|-----------|--------|------|---------|--------|
|------|-----------|--------|------|---------|--------|

unable to recognize "": no matches for kind "Ingress" in version "extensions/v1beta1"

В процессе установки используя следующую команду:

```
microk8s helm3 upgrade --install elma365 ./elma365 -f values-elma365.yaml --timeout=30m
```

Может возникнуть следующая ошибка:

```
Error: unable to build kubernetes objects from release manifest: unable to recognize "": no matches for kind "Ingress" in version "extensions/v1beta1"
```

Она связана с некорректной версией microk8s. Сначала нужно удалить microk8s командой:

```
snap remove microk8s
```

И установить корректную (на текущий момент):

```
snap install microk8s --classic --channel=1.32/stable
```

Получение конфигурации для 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:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCRENDQWV5Z0F3SUJBZ0lKQVBQMghoaFduTTR0TUEwR0NTcUdTSW
IzRFFQkN3VUFNQmN4RlRBVEJnTlYkQkFNTURERXdmakUxTWk0eE9ETXVNVFElRncweE9URXdNVFl3T1RRNE5UaGFgGdzAw
TnpBek1ETXdPVFE0TlRoYQpNQMn4RlRBVEJnTlZCQU1NRERFd0xqRTFNaTR4T0RNdU1UQ0NBu0l3RFFZSkvWklodmNOQV
FFQkJRQRnZ0VQckFEQ0NBuW9DZ2dFQkFQbWpiZ3F6NWxTL3FlTldlcDduVE9NUS94NXp6MUI5bGR3dTJKZkVxbEtIRXVW
T3RZYU0KeVJMb0xtYmZBbUxTd2ZDSXIycWhyZ1UxTlFTd096WUVCeXpNZ01WaWtzejZlVjJ3Um10b3RxbXpXLzZHQT1S0
pmM1MvTURERHh2ZmQva1hrb2lpeTE5WCtPVENDTHFUTmJZRG45RkdRTjJwSVVzVWV5Z255LzJjbkpbD0xLTGtBCm04NHVT
UXBLYUJ1T3hzeFZBU3J6cURINK1uczN5Y0pyWXRqSG9BaDBTcWxFdHhKblpG0GJpTGx2Q2lRVkMyNE4KQ2IyMkpRMHVyQW
FRU1BIRmRsaGtKU1RBL2Y0VFp1VDN2N2tpaU9QZWpGT252M0lXYld3UXZRYU5ad0JmM0NQwozK0dIRXNSU1NPYkQ5Z2FG
0DhVTXNScEM1TFBaVm12eFFVc0NBd0VBQWF0VE1GRXdIUUVlEVlIiwT0JCWUUVGUElKmp6ZWhaYW1nS0xPSkQyVitiU3VMaH
Fju01C0EdBMVVKsXdRWU1CYUFGUElKanplaFphbWdLTE9KRdJWk2JTdUwKaHFju01B0EdBMVVKRXdfQi93UUZNQU1CQWY4
d0RRWUpLb1pJaHJzTkFRRUxCUUFEZ2dFQkFI0VlPaGRMT3RXKwpxbkhnajRYMjcwSWRFNDNlc3JRRENEUnY4UGpRVmVwYy
tNUEdrMU10cFNsckgvTkNnc2puMzZ2RmpFeUJzcXFJckh4QmhPdi9wR3NIUHDua2p3bXNSYkdwWkliQTdRcGNKSTZlaEM4
aFUwSmI2b3ZwSU9zZnR6SFF6SVA4NWRh0EsKWGptbVY2UzNLQnZtZlplKMU1WZzFuUFBGYnpBdVhoaXI0RmFvZ3AzL2lwNk
VFdWtYNUYydnQvVDdSaC9X0DFDNQphUFhpTDdSMTUxQm9aU3U2Y2ZUdGx2Y092cw9DSDFvaXlvWlJIV2dSeElPVTU2dXVu
RXlUdVNoMjJnbGVSSlAyck8xWk9rNEVBS3VNMEsra05SeEVvcXNmeFllyLzicENvbGVJRXLlUUhNzJJU0RrdlVWRmhRMk
xGZTJl1bmtvRDgKY2JZbVAVU1tZ009Ci0tLS0tRU5EIEENFUlRJRklDQVRFLS0tLS0K

    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: S05BcnhqaWpCNz1RaWtDZEIzNHEzVU5qKzZCaERUbWxjQ2d6VTFmcjkrQT0K
```

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
```

Удаление завершенных работ

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

Прямой вызов 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.

Удаление подов в статусе Error

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

Сертификаты

Сертификаты

Проверка сертификатов

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

Обновление сертификатов

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
```

Погасить все поды

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

Кастомные порты Ingress

```
kubectl 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"}]}'
```

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

Установка с нуля

Установка с нуля

1. Базовая

Если ранее стоял microk8s, то сносим его:

```
snap remove microk8s
```

Далее ставим свежее (на момент написания была 1.32.8):

```
snap install microk8s --classic --channel=1.32/stable
```

Для удобства делаем алиасы:

```
snap alias microk8s.kubectl kubectl
snap alias microk8s.helm3 helm
```

Смотрим статус что включено что нет - минимум надо:

```
root@e365n1:~# microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    dns                # (core) CoreDNS
    ha-cluster         # (core) Configure high availability on the current node
    helm               # (core) Helm - the package manager for Kubernetes
    helm3              # (core) Helm 3 - the package manager for Kubernetes
    hostpath-storage  # (core) Storage class; allocates storage from host directory
    ingress            # (core) Ingress controller for external access
    metrics-server     # (core) K8s Metrics Server for API access to service metrics
    rbac                # (core) Role-Based Access Control for authorisation
    storage            # (core) Alias to hostpath-storage add-on, deprecated
  disabled:
    cert-manager       # (core) Cloud native certificate management
    cis-hardening      # (core) Apply CIS K8s hardening
    community           # (core) The community addons repository
```

```
dashboard      # (core) The Kubernetes dashboard
gpu            # (core) Alias to nvidia add-on
host-access    # (core) Allow Pods connecting to Host services smoothly
kube-ovn       # (core) An advanced network fabric for Kubernetes
mayastor       # (core) OpenEBS MayaStor
metallb        # (core) Loadbalancer for your Kubernetes cluster
minio          # (core) MinIO object storage
nvidia         # (core) NVIDIA hardware (GPU and network) support
observability  # (core) A lightweight observability stack for logs, traces and
metrics
  prometheus   # (core) Prometheus operator for monitoring and logging
  registry     # (core) Private image registry exposed on localhost:32000
  rook-ceph    # (core) Distributed Ceph storage using Rook
```

metric-server лучше ставить свежий [отсюда](#)

Установка с нуля

2. Metric Server

Было замечено, что в базовой снап установке metric server не первой свежести, поэтому стоит ставить отсюда:

```
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
```

Более подробно смотрим [сюда](#)

3. Патчим ingress

Без этого показывает 404. ELMA365 размещена в namespace default:

```
#!/usr/bin/env bash

nodesCount=$(microk8s kubectl get nodes -l elma365.com/node=initial --no-headers 2>/dev/null |
wc -l)
if [[ $nodesCount -eq 0 ]]; then
    microk8s kubectl label node --all elma365.com/node=initial --overwrite=true
fi

while true
do
    # microk8s enable storage dns rbac ingress linkerd && break
    microk8s enable storage dns rbac ingress && break
    sleep 5
done

if [ "$ELMA365_INSTALL_MODE" != "upgrade" ]; then
    microk8s kubectl patch ds/nginx-ingress-microk8s-controller -n ingress --type='json' -
p='[{"op": "replace", "path": "/spec/template/spec/containers/0/args", "value":["/nginx-
ingress-controller","--configmap=$(POD_NAMESPACE)/nginx-load-balancer-microk8s-conf","--tcp-
services-configmap=$(POD_NAMESPACE)/nginx-ingress-tcp-microk8s-conf","--udp-services-
configmap=$(POD_NAMESPACE)/nginx-ingress-udp-microk8s-conf"," ","--publish-status-
address=127.0.0.1"]}]'

    sleep 15

    echo "Patching the ingress k8s version 1.21"
    microk8s kubectl patch role nginx-ingress-microk8s-role -n ingress --type='json' -p='[{"op":
"add", "path":"/rules/3", "value":{"apiGroups":[""],"resourceNames":["ingress-controller-
leader"],"resources":["configmaps"],"verbs":["create","update"]}]'
    microk8s kubectl patch configmap nginx-load-balancer-microk8s-conf -n ingress --type='merge'
-p='{"data":{"annotation-value-word-
blocklist":["load_module, lua_package, _by_lua, root, serviceaccount\"]}}'
```

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

sleep 15
fi

echo "Waiting for disk provisioner"
while true
do
    microk8s kubectl -n kube-system wait --for=condition=Ready --timeout=1200s pod -l k8s-
app=hostpath-provisioner > /dev/null 2>&1 && break
    sleep 5
done

microk8s status --wait-ready > /dev/null
```

Очистка от подов в статусе Terminating

```
for p in $(kubectl get pods | grep Terminating | awk '{print $1}'); do kubectl delete pod $p --grace-period=0 --force;done
```

Установка в LXC

1. Создать СТ

- Убрать крыжик Unprivileged container (если этого не сделать потом нужны будут танцы с бубном)

2. Не запускать СТ и в Options отметить Nesting и FUSE

3. В консоли PVE в каталоге /etc/pve/lxc найти конфиг созданной СТ и добавить следующее:

```
lxc.apparmor.profile: unconfined
lxc.cap.drop:
lxc.mount.auto: proc:rw sys:rw
lxc.mount.entry: /dev/fuse dev/fuse none bind,create=file 0 0
lxc.mount.entry: /sys/kernel/security sys/kernel/security none bind,create=file 0 0
```

4. Запустить СТ и добавить в crontab (crontab -e):

```
@reboot ln -s /dev/console /dev/kmsg
```

5. Выполнить апдейт и установку необходимого:

```
apt update && apt upgrade -y && apt install snapd squashfuse fuse sudo -y
```

6. Перезапустить СТ

7. Установить microk8s:

```
snap install microk8s --classic
```