Установка с нуля

- 1. Базовая
- 2. Metric Server
- 3. Патчим ingress

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

2. Metric Server

Было замечено, что в базовой snap 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
```