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

У меня есть кластер на k0s например, я хочу подцепить к нему какой-нибудь воркер, где угодно. Наброски команд.

По мотивам https://github.com/kelseyhightower/kubernetes-the-hard-way

```
###

root@kube-ctrl:~/add-manually-worker# cat ca.conf
[req]
distinguished_name = req_distinguished_name
prompt             = no
x509_extensions    = ca_x509_extensions

[ca_x509_extensions]
basicConstraints = CA:TRUE
keyUsage         = cRLSign, keyCertSign

[req_distinguished_name]
C   = US
ST  = Washington
L   = Seattle
CN  = CA

# Worker Nodes
#
# Kubernetes uses a [special-purpose authorization
mode](https://kubernetes.io/docs/admin/authorization/node/)
# called Node Authorizer, that specifically authorizes API requests made
# by [Kubelets](https://kubernetes.io/docs/concepts/overview/components/#kubelet).
# In order to be authorized by the Node Authorizer, Kubelets must use a credential
# that identifies them as being in the `system:nodes` group, with a username
# of `system:node:<nodeName>`.

[kube-work-4]
distinguished_name = kube-work-4_distinguished_name
prompt             = no
```

```
req_extensions     = kube-work-4_req_extensions

[kube-work-4_req_extensions]
basicConstraints     = CA:FALSE
extendedKeyUsage     = clientAuth, serverAuth
keyUsage             = critical, digitalSignature, keyEncipherment
nsCertType           = client
nsComment            = "kube-work-4 Certificate"
subjectAltName       = DNS:kube-work-4, IP:10.138.117.7 # IP воркера
subjectKeyIdentifier = hash

[kube-work-4_distinguished_name]
CN = system:node:kube-work-4
O  = system:nodes
C  = US
ST = Washington
L  = Seattle

[default_req_extensions]
basicConstraints     = CA:FALSE
extendedKeyUsage     = clientAuth
keyUsage             = critical, digitalSignature, keyEncipherment
nsCertType           = client
nsComment            = "Admin Client Certificate"
subjectKeyIdentifier = hash




###
for host in kube-work-4; do
  openssl genrsa -out "${host}.key" 4096

  openssl req -new -key "${host}.key" -sha256 \
    -config "ca.conf" -section ${host} \
    -out "${host}.csr"

  openssl x509 -req -days 3653 -in "${host}.csr" \
    -copy_extensions copyall \
    -sha256 -CA "ca.crt" \
    -CAkey "ca.key" \
```

```
    -CAcreateserial \
    -out "${host}.crt"

  kubectl config set-cluster kubernetes-the-hard-way \
    --certificate-authority=ca.crt \
    --embed-certs=true \
    --server=https://10.138.117.204:6443 \
    --kubeconfig=${host}.kubeconfig
done

###

root@kube-ctrl:~/add-manually-worker# cat kube-work-4.kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: `cat ca.crt | base64 -w0`
    server: https://10.138.117.204:6443
  name: kubernetes-the-hard-way
contexts:
- context:
    cluster: kubernetes-the-hard-way
    namespace: default
    user: system:node:kube-work-4
  name: default-context
current-context: default-context
kind: Config
preferences: {}
users:
- name: system:node:kube-work-4
  user:
    client-certificate-data: `cat crt | base64 -w0`
    client-key-data: `cat key | base64 -w0`

###

root@kube-ctrl:~/add-manually-worker# kubectl --kubeconfig kube-work-4.kubeconfig get no
NAME          STATUS   ROLES    AGE   VERSION
kube-work-1   Ready    <none>   85d   v1.30.3+k0s
kube-work-2   Ready    shits    15h   v1.30.6+k0s
```

```
kube-work-3   Ready    <none>   14h   v1.30.5+k0s

###

root@kube-work-4:~# cat down
https://github.com/containernetworking/plugins/releases/download/v1.3.0/cni-plugins-linux-amd64-v1.3.0.tgz
https://storage.googleapis.com/kubernetes-release/release/v1.30.1/bin/linux/amd64/kube-proxy
https://storage.googleapis.com/kubernetes-release/release/v1.30.1/bin/linux/amd64/kubelet

root@kube-work-4:~# wget -q --show-progress   --https-only   --timestamping   -P downloads   -i down
cni-plugins-linux-amd64-v1.3.0.tgz
100%[===================================================================================================>]  43.24M  21.5MB/s    in 2.0s
kube-proxy
100%[===================================================================================================>]  54.91M  17.3MB/s    in 3.2s
kubelet
100%[===================================================================================================>]  95.46M  22.5MB/s    in 5.0s
apt install containerd -y

###

root@kube-work-4:~# systemctl cat kube.service
# /etc/systemd/system/kube.service
[Unit]
Description=Kubernetes Kubelet
Documentation=https://github.com/kubernetes/kubernetes
After=containerd.service
Requires=containerd.service

[Service]
ExecStart=/usr/local/bin/kubelet \
  --config=/var/lib/kubelet/kubelet-config.yaml \
  --kubeconfig=/var/lib/kubelet/kubeconfig \
  --register-node=true \
  --v=2
Restart=on-failure
```

```
RestartSec=5

[Install]
WantedBy=multi-user.target

###

root@kube-work-4:~# cat /var/lib/kubelet/kubelet-config.yaml
kind: KubeletConfiguration
apiVersion: kubelet.config.k8s.io/v1beta1
authentication:
  anonymous:
    enabled: false
  webhook:
    enabled: true
  x509:
    clientCAFile: "/var/lib/kubelet/ca.crt"
authorization:
  mode: Webhook
clusterDomain: "cluster.local"
clusterDNS:
  - "10.32.0.10"
cgroupDriver: systemd
containerRuntimeEndpoint: "unix:///var/run/containerd/containerd.sock"
podCIDR: "100.64.32.0/24"
resolvConf: "/etc/resolv.conf"
runtimeRequestTimeout: "15m"
tlsCertFile: "/var/lib/kubelet/kubelet.crt"
tlsPrivateKeyFile: "/var/lib/kubelet/kubelet.key"

###
kube-worker-4.{crt,key}->/var/lib/kubelet/kubelet.{crt,key}

###

root@kube-work-4:~# cat /var/lib/kubelet/kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: # ca base64
```

```
    server: https://10.138.117.204:6443
  name: kubernetes-the-hard-way
contexts:
- context:
    cluster: kubernetes-the-hard-way
    namespace: default
    user: system:node:kube-work-4
  name: default-context
current-context: default-context
kind: Config
preferences: {}
users:
- name: system:node:kube-work-4
  user:
    client-certificate-data: # base64
    client-key-data: # base64


###

systemctl start kube.service

root@kube-ctrl:~# kubectl get no -o wide
NAME          STATUS   ROLES    AGE   VERSION       INTERNAL-IP      EXTERNAL-IP   OS-
IMAGE                         KERNEL-VERSION                CONTAINER-RUNTIME
kube-work-1   Ready    <none>   85d   v1.30.3+k0s   10.138.117.150   <none>        Debian
GNU/Linux 12 (bookworm)   6.1.0-23-amd64                containerd://1.7.20
kube-work-2   Ready    shits    16h   v1.30.6+k0s   10.13.37.3       <none>        Rocky Linux
9.4 (Blue Onyx)      5.14.0-427.42.1.el9_4.x86_64   containerd://1.7.22
kube-work-3   Ready    <none>   14h   v1.30.5+k0s   10.138.117.79    <none>        AlmaLinux
9.4 (Seafoam Ocelot)   5.14.0-427.42.1.el9_4.x86_64   containerd://1.7.22
kube-work-4   Ready    <none>   35s   v1.30.1       10.138.117.7     <none>        Ubuntu
20.04.6 LTS               5.4.0-200-generic             containerd://1.7.12
```

Revision #2
Created 2024-11-09 12:42:55 MSK by Ivan
Updated 2024-11-09 15:20:56 MSK by Ivan