

Nastavení kubectl

- [Postup](#)
- [Instalace](#)
- [Konfigurace](#)
 - [Připojení ke clusteru z masteru s existující konfigurací \(např. po instalaci clusteru\)](#)
 - [Vytvoření nové konfigurace](#)
 - [Vytvoření uživatele \(service account\) new-custom-user](#)
 - [Přiřazení práv pro nového uživatele](#)
 - [Získání příslušných údajů](#)
 - [Generování configu](#)
- [Quality of life tuning](#)
 - [Alias](#)
 - [Textový editor](#)
 - [Config contexty - WIP](#)

Postup

Postup je prováděn z uzlu, na němž má kubectl běžet.

Instalace

Instalace kubectl

```
dnf install -y kubectl
```

Konfigurace

Konfigurace kubectl lze upravovat skrz config soubor. Tento soubor je dodán do kubectl následujícími způsoby v uvedeném pořadí (předcházející způsob má přednost):

- `--kubeconfig <config_location>` - parametr kubectl
- `KUBECONFIG = <config_location>` - proměnná prostředí
- default filesystem location - `$HOME/.kube/config` - **doporučený způsob**

Připojení ke clusteru z masteru s existující konfigurací (např. po instalaci clusteru)

Při inicializaci nové clusteru, či připojení master nódu ke stávajícímu clusteru, je na nódu automaticky nastavena konfigurace pro administrátora do `/etc/kubernetes/admin.conf`

Tuto konfiguraci lze zkopírovat do výchozího adresáře pro kubectl, aby kubectl se dokázalo ke clusteru přihlásit.

kubectl config

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Vytvoření nové konfigurace

Vytvoření nové konfigurace je prováděno na uzlu s již fungujícím kubectl (např. masteru). Návod popisuje vytvoření uživatele s cluster-admin rolí.

Vytvoření uživatele (service account) new-custom-user

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: new-custom-user
  namespace: kube-system
automountServiceAccountToken: true
```

Přřazení práv pro nového uživatele

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: kommander-cluster-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: new-custom-user
  namespace: kube-system
```

Získání příslušných údajů

```
export USER_TOKEN_NAME=$(kubectl -n kube-system get serviceaccount kommander-cluster-admin -o=jsonpath='{.secrets[0].name}')
export USER_TOKEN_VALUE=$(kubectl -n kube-system get secret/${USER_TOKEN_NAME} -o=go-template='{{.data.token}}' | base64 --decode)
export CURRENT_CONTEXT=$(kubectl config current-context)
export CURRENT_CLUSTER=$(kubectl config view --raw -o=go-template='{{range .contexts}}{{if eq .name "'${CURRENT_CONTEXT}"'}}{{ index .context "cluster" }}{{end}}{{end}}')
export CLUSTER_CA=$(kubectl config view --raw -o=go-template='{{range .clusters}}{{if eq .name "'${CURRENT_CLUSTER}"'}}{{with index .cluster "certificate-authority-data" }}{{.}}{{end}}{{ end }}{{ end }}')
export CLUSTER_SERVER=$(kubectl config view --raw -o=go-template='{{range .clusters}}{{if eq .name "'${CURRENT_CLUSTER}"'}}{{ .cluster.server }}{{end}}{{ end }}')
```

Generování configu

```
apiVersion: v1
kind: Config
current-context: ${CURRENT_CONTEXT}
contexts:
- name: ${CURRENT_CONTEXT}
  context:
    cluster: ${CURRENT_CONTEXT}
    user: kommander-cluster-admin
    namespace: kube-system
clusters:
- name: ${CURRENT_CONTEXT}
  cluster:
    certificate-authority-data: ${CLUSTER_CA}
    server: ${CLUSTER_SERVER}
users:
- name: kommander-cluster-admin
  user:
    token: ${USER_TOKEN_VALUE}
```

Quality of life tuning

Alias

Je dobrým zvykem nastavit alias k pro kubectl

kubectl alias

```
alias k=kubectl
```

Pro persistenci tohoto nastavení i přes restarty jej lze vložit mezi skripty spouštěné při nové terminálové session, např. `/etc/profile.d/kubectl.sh`

Textový editor

Pro změnu výchozího nástroje pro úpravu resources (např. přes `kubectl edit ...`) lze nastavit proměnnou prostředí

```
KUBE_EDITOR=nano
```

Config contexty - WIP

V konfiguraci kubectl lze mít více nastavení zároveň, mezi těmito lze přepínat přes

kubectl config

```
kubectl config get-contexts  
kubectl config set-contexts
```