# Usage

# kubectrl

## Fish shell completion

```
kubectl completion fish > /usr/share/fish/vendor_completions.d/kubectl.fish
source /usr/share/fish/vendor_completions.d/kubectl.fish
```

## Flags

```
kubectl __complete -
--as Username to impersonate for the operation. User could be a regular user or a service
account in a namespace.
--as-group Group to impersonate for the operation, this flag can be repeated to specify
multiple groups.
--as-uid UID to impersonate for the operation.
--cache-dir Default cache directory
--certificate-authority Path to a cert file for the certificate authority
--client-certificate Path to a client certificate file for TLS
--client-key Path to a client key file for TLS
--cluster The name of the kubeconfig cluster to use
--context The name of the kubeconfig context to use
--disable-compression If true, opt-out of response compression for all requests to the server
--help help for kubectl
-h help for kubectl
--insecure-skip-tls-verify If true, the server's certificate will not be checked for validity.
This will make your HTTPS connections insecure
--kubeconfig Path to the kubeconfig file to use for CLI requests.
--log-flush-frequency Maximum number of seconds between log flushes
--match-server-version Require server version to match client version
--namespace If present, the namespace scope for this CLI request
-n If present, the namespace scope for this CLI request
--password Password for basic authentication to the API server
```

```
--profile Name of profile to capture. One of (none|cpu|heap|goroutine|threadcreate|block|mutex)
--profile-output Name of the file to write the profile to
--request-timeout The length of time to wait before giving up on a single server request. Non-zero values should contain a corresponding time unit (e.g. 1s, 2m, 3h). A value of zero means don't timeout requests.
--server The address and port of the Kubernetes API server
-s The address and port of the Kubernetes API server
--tls-server-name Server name to use for server certificate validation. If it is not provided, the hostname used to contact the server is used
--token Bearer token for authentication to the API server
--user The name of the kubeconfig user to use
--username Username for basic authentication to the API server
--v number for the log level verbosity
-v number for the log level verbosity
--vmodule comma-separated list of pattern=N settings for file-filtered logging (only works for the default text log format)
--warnings-as-errors Treat warnings received from the server as errors and exit with a non-zero exit code
:4
```

# Server and certs

```
kubectl --server https://kube-m0:6443/ --certificate-authority /etc/kubernetes/pki/ca.crt get nodes
```

# Authentication with token

## Get token (as root)

```
kubeadm token create
```

## Use token

```
kubectl --server https://kube-m0:6443/ --certificate-authority /etc/kubernetes/pki/ca.crt --token <token> cluster-info dump
```

# Authentication with certificate

## Requesting certificate signing

- https://kubernetes.io/docs/reference/access-authn-authz/certificate-signing-requests/#normal-user

**mksignreq**

```
#!/bin/sh

USR="$1"
CSR=`cat "certs/$USR.csr" | base64 -w 0`

cat <<EOF
apiVersion: certificates.k8s.io/v1
kind: CertificateSigningRequest
metadata:
  name: $USR
spec:
  request: $CSR
  signerName: kubernetes.io/kube-apiserver-client
  expirationSeconds: 2147483647  # max
  usages:
  - client auth
EOF
```

```
./mksignreq hxd | kubectl --server https://kube-m0:6443/ --certificate-authority
/etc/kubernetes/pki/ca.crt --token $TOKEN apply -f -
kubectl --server https://kube-m0:6443/ --certificate-authority /etc/kubernetes/pki/ca.crt --
token $TOKEN certificate approve hxd
```

Get the cert (requires `kube-kontroller-manager` running as it is responsible for signing):

```
kubectl --server https://kube-m0:6443/ --certificate-authority /etc/kubernetes/pki/ca.crt
 --token $TOKEN get csr hxd -o jsonpath='{.status.certificate}' | base64 -d > certs/hxd.crt
```

# Use certificate

```
kubectl --server https://kube-m0:6443/ --certificate-authority /etc/kubernetes/pki/ca.crt
 --client-key certs/hxd.key --client-certificate certs/hxd.crt cluster-info dump
```

# Generate configuration

```
kubectl config set-cluster kubernetes --server https://kube-m0:6443/ --certificate-author
ity /etc/kubernetes/pki/ca.crt
kubectl config set-credentials hxd --client-key certs/hxd.key --client-certificate
certs/hxd.crt
kubectl config set-context hxd@kubernetes --cluster=kubernetes --user=hxd
kubectl config set current-context hxd@kubernetes
```

This should generate file `.kube/config`:

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /etc/kubernetes/pki/ca.crt
    server: https://kube-m0:6443/
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: hxd
  name: hxd@kubernetes
current-context: hxd@kubernetes
kind: Config
preferences: {}
users:
- name: hxd
  user:
    client-certificate: /home/hxd/certs/hxd.crt
    client-key: /home/hxd/certs/hxd.key
```

# Terraform

- https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs

```
provider "kubernetes" {
  config_path    = "~/.kube/config"
}
```

```
resource "kubernetes_namespace" "example" {
  metadata {
    name = "my-first-namespace"
  }
}
```

```
terraform init
terraform plan
terraform apply
```

---

Revision #12
Created 2023-04-24 18:30:55 IST by hxd
Updated 2023-11-11 11:45:20 GMT by hxd