Kubernetes - Kubectl: The Missing Manpages

The kubectl CLI is written in Golang, and uses Cobra to automatically generate documentation. You use 'kubectl help' and 'kubectl [COMMAND] --help' to get manpage-like output. Since these aren't available as manpages, they won't be found in various WWW manpage collections to read (so you can look at them on a device without Kubernetes installed like a phone). Not included here is output of 'kubectl explain' for anything, so it's just 'the missing manpages'

==========

```
kubectl help
kubectl controls the Kubernetes cluster manager.

 Find more information at: https://kubernetes.io/docs/reference/kubectl/

Basic Commands (Beginner):
  create          Create a resource from a file or from stdin
  expose          Take a replication controller, service, deployment or pod andexpose it as
a new Kubernetes service
  run             Run a particular image on the cluster
  set             Set specific features on objects

Basic Commands (Intermediate):
  explain         Get documentation for a resource
  get             Display one or many resources
  edit            Edit a resource on the server
  delete          Delete resources by file names, stdin, resources and names, or by
resources and label selector

Deploy Commands:
  rollout         Manage the rollout of a resource
  scale           Set a new size for a deployment, replica set, or replication controller
  autoscale       Auto-scale a deployment, replica set, stateful set, or replication
controller

Cluster Management Commands:
  certificate     Modify certificate resources.
  cluster-info    Display cluster information
  top             Display resource (CPU/memory) usage
  cordon          Mark node as unschedulable
  uncordon        Mark node as schedulable
  drain           Drain node in preparation for maintenance
  taint           Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe        Show details of a specific resource or group of resources
  logs            Print the logs for a container in a pod
  attach          Attach to a running container
  exec            Execute a command in a container
  port-forward    Forward one or more local ports to a pod
  proxy           Run a proxy to the Kubernetes API server
  cp              Copy files and directories to and from containers
  auth            Inspect authorization
  debug           Create debugging sessions for troubleshooting workloads and nodes
  events          List events

Advanced Commands:
  diff            Diff the live version against a would-be applied version
  apply           Apply a configuration to a resource by file name or stdin
  patch           Update fields of a resource
  replace         Replace a resource by file name or stdin
  wait            Experimental: Wait for a specific condition on one or many resources
  kustomize       Build a kustomization target from a directory or URL.

Settings Commands:
  label           Update the labels on a resource
  annotate        Update the annotations on a resource
```

```
  completion      Output shell completion code for the specified shell (bash, zsh, fish, or
powershell)

Other Commands:
  alpha           Commands for features in alpha
  api-resources   Print the supported API resources on the server
  api-versions    Print the supported API versions on the server, in the form of
"group/version"
  config          Modify kubeconfig files
  plugin          Provides utilities for interacting with plugins
  version         Print the client and server version information

Usage:
  kubectl [flags] [options]

  ================================================================================


[tm@freeipa ~]$ kubectl options
The following options can be passed to any command:

    --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='/home/tm/.kube/cache':
       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=false:
       If true, opt-out of response compression for all requests to the server

    --insecure-skip-tls-verify=false:
       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=5s:
       Maximum number of seconds between log flushes

    --match-server-version=false:
       Require server version to match client version
```

```
    -n, --namespace='':
       If present, the namespace scope for this CLI request

    --password='':
       Password for basic authentication to the API server

    --profile='none':
       Name of profile to capture. One of (none|cpu|heap|goroutine|threadcreate|block|mutex)

    --profile-output='profile.pprof':
       Name of the file to write the profile to

    --request-timeout='0':
       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.

    -s, --server='':
       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, --v=0:
       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=false:
       Treat warnings received from the server as errors and exit with a non-zero exit code
```

```
  ================================================================================

kubectl create --help
Create a resource from a file or from stdin.

 JSON and YAML formats are accepted.

Examples:
  # Create a pod using the data in pod.json
  kubectl create -f ./pod.json

  # Create a pod based on the JSON passed into stdin
  cat pod.json | kubectl create -f -
```

```
  # Edit the data in registry.yaml in JSON then create the resource using the edited data
  kubectl create -f registry.yaml --edit -o json

Available Commands:
  clusterrole           Create a cluster role
  clusterrolebinding    Create a cluster role binding for a particular cluster role
  configmap             Create a config map from a local file, directory or literal value
  cronjob               Create a cron job with the specified name
  deployment            Create a deployment with the specified name
  ingress               Create an ingress with the specified name
  job                   Create a job with the specified name
  namespace             Create a namespace with the specified name
  poddisruptionbudget   Create a pod disruption budget with the specified name
  priorityclass         Create a priority class with the specified name
  quota                 Create a quota with the specified name
  role                  Create a role with single rule
  rolebinding           Create a role binding for a particular role or cluster role
  secret                Create a secret using specified subcommand
  service               Create a service using a specified subcommand
  serviceaccount        Create a service account with the specified name
  token                 Request a service account token

Options:
    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    --edit=false:
       Edit the API resource before creating

    --field-manager='kubectl-create':
       Name of the manager used to track field ownership.

    -f, --filename=[ ]:
       Filename, directory, or URL to files to use to create the resource

    -k, --kustomize='':
       Process the kustomization directory. This flag can't be used together
       with -f or -R.

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file).

    --raw='':
       Raw URI to POST to the server.  Uses the transport specified by the
       kubeconfig file.

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    --save-config=false:
       If true, the configuration of current object will be saved in its
       annotation. Otherwise, the annotation will be unchanged. This flag is
       useful when you want to perform kubectl apply on this object in the
       future.
```

```
    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    --validate='strict':
       Must be one of: strict (or true), warn, ignore (or false).
             "true" or "strict" will use a schema to validate the input and fail the request
       if invalid. It will perform server side validation if
       ServerSideFieldValidation is enabled on the api-server, but will fall
       back to less reliable client-side validation if not.
       "warn" will
       warn about unknown or duplicate fields without blocking the request if
       server-side field validation is enabled on the API server, and behave
       as "ignore" otherwise.
             "false" or "ignore" will not perform any
       schema validation, silently dropping any unknown or duplicate fields.

    --windows-line-endings=false:
       Only relevant if --edit=true. Defaults to the line ending native to
       your platform.

Usage:
  kubectl create -f FILENAME [options]



================================================================================

kubectl expose --help

Expose a resource as a new Kubernetes service.

 Looks up a deployment, service, replica set, replication controller or pod by
name and uses the selector for that resource as the selector for a new service
on the specified port. A deployment or replica set will be exposed as a service
only if its selector is convertible to a selector that service supports, i.e.
when the selector contains only the matchLabels component. Note that if no port
is specified via --port and the exposed resource has multiple ports, all will be
re-used by the new service. Also if no labels are specified, the new service
will re-use the labels from the resource it exposes.

 Possible resources include (case insensitive):

 pod (po), service (svc), replicationcontroller (rc), deployment (deploy),
replicaset (rs)

Examples:
  # Create a service for a replicated nginx, which serves on port 80 and connects to the
containers on port 8000
  kubectl expose rc nginx --port=80 --target-port=8000

  # Create a service for a replication controller identified by type and name specified in
"nginx-controller.yaml", which serves on port 80 and connects tothe containers on port 8000
  kubectl expose -f nginx-controller.yaml --port=80 --target-port=8000

  # Create a service for a pod valid-pod, which serves on port 444 with the name "frontend"
```

```
  kubectl expose pod valid-pod --port=444 --name=frontend

  # Create a second service based on the above service, exposing the container port 8443 as
port 443 with the name "nginx-https"
  kubectl expose service nginx --port=443 --target-port=8443 --name=nginx-https

  # Create a service for a replicated streaming application on port 4100 balancing UDP
traffic and named 'video-stream'.
  kubectl expose rc streamer --port=4100 --protocol=UDP --name=video-stream

  # Create a service for a replicated nginx using replica set, which serves on port 80 and
connects to the containers on port 8000
  kubectl expose rs nginx --port=80 --target-port=8000

  # Create a service for an nginx deployment, which serves on port 80 and connects to the
containers on port 8000
  kubectl expose deployment nginx --port=80 --target-port=8000

Options:
    --allow-missing-template-keys=true:
      If true, ignore any errors in templates when a field or map key is
      missing in the template. Only applies to golang and jsonpath output
      formats.

    --cluster-ip='':
      ClusterIP to be assigned to the service. Leave empty to auto-allocate,
      or set to 'None' to create a headless service.

    --dry-run='none':
      Must be "none", "server", or "client". If client strategy, only print
      the object that would be sent, without sending it. If server strategy,
      submit server-side request without persisting the resource.

    --external-ip='':
      Additional external IP address (not managed by Kubernetes) to accept
      for the service. If this IP is routed to a node, the service can be
      accessed by this IP in addition to its generated service IP.

    --field-manager='kubectl-expose':
      Name of the manager used to track field ownership.

    -f, --filename=[ ]:
      Filename, directory, or URL to files identifying the resource to
      expose a service

    -k, --kustomize='':
      Process the kustomization directory. This flag can't be used together
      with -f or -R.

    -l, --labels='':
      Labels to apply to the service created by this call.

    --load-balancer-ip='':
      IP to assign to the LoadBalancer. If empty, an ephemeral IP will be
      created and used (cloud-provider specific).

    --name='':
      The name for the newly created object.

    -o, --output='':
      Output format. One of: (json, yaml, name, go-template,
      go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
      jsonpath-file).

    --override-type='merge':
      The method used to override the generated object: json, merge, or
```

```
       strategic.

    --overrides='':
       An inline JSON override for the generated object. If this is
       non-empty, it is used to override the generated object. Requires that
       the object supply a valid apiVersion field.

    --port='':
       The port that the service should serve on. Copied from the resource
       being exposed, if unspecified

    --protocol='':
       The network protocol for the service to be created. Default is 'TCP'.

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    --save-config=false:
       If true, the configuration of current object will be saved in its
       annotation. Otherwise, the annotation will be unchanged. This flag is
       useful when you want to perform kubectl apply on this object in the
       future.

    --selector='':
       A label selector to use for this service. Only equality-based selector
       requirements are supported. If empty (the default) infer the selector
       from the replication controller or replica set.)

    --session-affinity='':
       If non-empty, set the session affinity for the service to this; legal
       values: 'None', 'ClientIP'

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --target-port='':
       Name or number for the port on the container that the service should
       direct traffic to. Optional.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    --type='':
       Type for this service: ClusterIP, NodePort, LoadBalancer, or
       ExternalName. Default is 'ClusterIP'.

Usage:
  kubectl expose (-f FILENAME | TYPE NAME) [--port=port] [--protocol=TCP|UDP|SCTP] [--
target-port=number-or-name] [--name=name] [--external-ip=external-ip-of-service] [--
type=type] [options]


================================================================================
kubectl run --help
Create and run a particular image in a pod.

Examples:
  # Start a nginx pod
  kubectl run nginx --image=nginx

  # Start a hazelcast pod and let the container expose port 5701
```

```
  kubectl run hazelcast --image=hazelcast/hazelcast --port=5701

  # Start a hazelcast pod and set environment variables "DNS_DOMAIN=cluster" and
"POD_NAMESPACE=default" in the container
  kubectl run hazelcast --image=hazelcast/hazelcast --env="DNS_DOMAIN=cluster" --
env="POD_NAMESPACE=default"

  # Start a hazelcast pod and set labels "app=hazelcast" and "env=prod" in the container
  kubectl run hazelcast --image=hazelcast/hazelcast --labels="app=hazelcast,env=prod"

  # Dry run; print the corresponding API objects without creating them
  kubectl run nginx --image=nginx --dry-run=client

  # Start a nginx pod, but overload the spec with a partial set of values parsed from JSON
  kubectl run nginx --image=nginx --overrides='{ "apiVersion": "v1", "spec": { ... } }'

  # Start a busybox pod and keep it in the foreground, don't restart it if it exits
  kubectl run -i -t busybox --image=busybox --restart=Never

  # Start the nginx pod using the default command, but use custom arguments (arg1 .. argN)
for that command
  kubectl run nginx --image=nginx -- <arg1> <arg2> ... <argN>

  # Start the nginx pod using a different command and custom arguments
  kubectl run nginx --image=nginx --command -- <cmd> <arg1> ... <argN>

Options:
    --allow-missing-template-keys=true:
      If true, ignore any errors in templates when a field or map key is
      missing in the template. Only applies to golang and jsonpath output
      formats.

    --annotations=[ ]:
      Annotations to apply to the pod.

    --attach=false:
      If true, wait for the Pod to start running, and then attach to the Pod
      as if 'kubectl attach ...' were called.  Default false, unless
      '-i/--stdin' is set, in which case the default is true. With
      '--restart=Never' the exit code of the container process is returned.

    --command=false:
      If true and extra arguments are present, use them as the 'command'
      field in the container, rather than the 'args' field which is the
      default.

    --dry-run='none':
      Must be "none", "server", or "client". If client strategy, only print
      the object that would be sent, without sending it. If server strategy,
      submit server-side request without persisting the resource.

    --env=[ ]:
      Environment variables to set in the container.

    --expose=false:
      If true, create a ClusterIP service associated with the pod.  Requires
      `--port`.

    --field-manager='kubectl-run':
      Name of the manager used to track field ownership.

    --image='':
      The image for the container to run.

    --image-pull-policy='':
      The image pull policy for the container.  If left empty, this value
```

```
   will not be specified by the client and defaulted by the server.

-l, --labels='':
   Comma separated labels to apply to the pod. Will override previous
   values.

--leave-stdin-open=false:
   If the pod is started in interactive mode or with stdin, leave stdin
   open after the first attach completes. By default, stdin will be
   closed after the first attach completes.

-o, --output='':
   Output format. One of: (json, yaml, name, go-template,
   go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
   jsonpath-file).

--override-type='merge':
   The method used to override the generated object: json, merge, or
   strategic.

--overrides='':
   An inline JSON override for the generated object. If this is
   non-empty, it is used to override the generated object. Requires that
   the object supply a valid apiVersion field.

--pod-running-timeout=1m0s:
   The length of time (like 5s, 2m, or 3h, higher than zero) to wait
   until at least one pod is running

--port='':
   The port that this container exposes.

--privileged=false:
   If true, run the container in privileged mode.

-q, --quiet=false:
   If true, suppress prompt messages.

--restart='Always':
   The restart policy for this Pod.  Legal values [Always, OnFailure,
   Never].

--rm=false:
   If true, delete the pod after it exits.  Only valid when attaching to
   the container, e.g. with '--attach' or with '-i/--stdin'.

--save-config=false:
   If true, the configuration of current object will be saved in its
   annotation. Otherwise, the annotation will be unchanged. This flag is
   useful when you want to perform kubectl apply on this object in the
   future.

--show-managed-fields=false:
   If true, keep the managedFields when printing objects in JSON or YAML
   format.

-i, --stdin=false:
   Keep stdin open on the container in the pod, even if nothing is
   attached.

--template='':
   Template string or path to template file to use when -o=go-template,
   -o=go-template-file. The template format is golang templates
   [http://golang.org/pkg/text/template/#pkg-overview].

-t, --tty=false:
```

```
      Allocate a TTY for the container in the pod.

Usage:
  kubectl run NAME --image=image [--env="key=value"] [--port=port] [--dry-run=server|client]
[--overrides=inline-json] [--command] -- [COMMAND] [args...] [options]


================================================================================
kubectl set --help
Configure application resources.

 These commands help you make changes to existing application resources.

Available Commands:
  env               Update environment variables on a pod template
  image             Update the image of a pod template
  resources         Update resource requests/limits on objects with pod templates
  selector          Set the selector on a resource
  serviceaccount    Update the service account of a resource
  subject           Update the user, group, or service account in a role binding or cluster
role binding

Usage:
  kubectl set SUBCOMMAND [options]


================================================================================
kubectl get --help
Display one or many resources.

 Prints a table of the most important information about the specified resources.
You can filter the list using a label selector and the --selector flag. If the
desired resource type is namespaced you will only see results in your current
namespace unless you pass --all-namespaces.

 By specifying the output as 'template' and providing a Go template as the value
of the --template flag, you can filter the attributes of the fetched resources.

Use "kubectl api-resources" for a complete list of supported resources.

Examples:
  # List all pods in ps output format
  kubectl get pods

  # List all pods in ps output format with more information (such as node name)
  kubectl get pods -o wide

  # List a single replication controller with specified NAME in ps output format
  kubectl get replicationcontroller web

  # List deployments in JSON output format, in the "v1" version of the "apps" API group
  kubectl get deployments.v1.apps -o json

  # List a single pod in JSON output format
  kubectl get -o json pod web-pod-13je7

  # List a pod identified by type and name specified in "pod.yaml" in JSON output format
  kubectl get -f pod.yaml -o json

  # List resources from a directory with kustomization.yaml - e.g. dir/kustomization.yaml
  kubectl get -k dir/

  # Return only the phase value of the specified pod
  kubectl get -o template pod/web-pod-13je7 --template={{.status.phase}}

  # List resource information in custom columns
```

```
  kubectl get pod test-pod -o custom-
columns=CONTAINER:.spec.containers[0].name,IMAGE:.spec.containers[0].image

  # List all replication controllers and services together in ps output format
  kubectl get rc,services

  # List one or more resources by their type and names
  kubectl get rc/web service/frontend pods/web-pod-13je7

  # List status subresource for a single pod.
  kubectl get pod web-pod-13je7 --subresource status

Options:
    -A, --all-namespaces=false:
       If present, list the requested object(s) across all namespaces.
       Namespace in current context is ignored even if specified with
       --namespace.

    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --chunk-size=500:
       Return large lists in chunks rather than all at once. Pass 0 to
       disable. This flag is beta and may change in the future.

    --field-selector='':
       Selector (field query) to filter on, supports '=', '==', and
       '!='.(e.g. --field-selector key1=value1,key2=value2). The server only
       supports a limited number of field queries per type.

    -f, --filename=[ ]:
       Filename, directory, or URL to files identifying the resource to get
       from a server.

    --ignore-not-found=false:
       If the requested object does not exist the command will return exit
       code 0.

    -k, --kustomize='':
       Process the kustomization directory. This flag can't be used together
       with -f or -R.

    -L, --label-columns=[ ]:
       Accepts a comma separated list of labels that are going to be
       presented as columns. Names are case-sensitive. You can also use
       multiple flag options like -L label1 -L label2...

    --no-headers=false:
       When using the default or custom-column output format, don't print
       headers (default print headers).

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file, custom-columns, custom-columns-file, wide). See custom
       columns
       [https://kubernetes.io/docs/reference/kubectl/#custom-columns], golang
       template [http://golang.org/pkg/text/template/#pkg-overview] and
       jsonpath template
       [https://kubernetes.io/docs/reference/kubectl/jsonpath/].

    --output-watch-events=false:
       Output watch event objects when --watch or --watch-only is used.
       Existing objects are output as initial ADDED events.
```

```
    --raw='':
       Raw URI to request from the server.  Uses the transport specified by
       the kubeconfig file.

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

    --server-print=true:
       If true, have the server return the appropriate table output. Supports
       extension APIs and CRDs.

    --show-kind=false:
       If present, list the resource type for the requested object(s).

    --show-labels=false:
       When printing, show all labels as the last column (default hide labels
       column)

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --sort-by='':
       If non-empty, sort list types using this field specification.  The
       field specification is expressed as a JSONPath expression (e.g.
       '{.metadata.name}'). The field in the API resource specified by this
       JSONPath expression must be an integer or a string.

    --subresource='':
       If specified, gets the subresource of the requested object. Must be
       one of [status scale]. This flag is alpha and may change in the
       future.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    -w, --watch=false:
       After listing/getting the requested object, watch for changes.

    --watch-only=false:
       Watch for changes to the requested object(s), without listing/getting
       first.

Usage:
  kubectl get
[(-o|--output=)json|yaml|name|go-template|go-template-file|template|templatefile|jsonpath|
jsonpath-as-json|jsonpath-file|custom-columns|custom-columns-file|wide] (TYPE[.VERSION]
[.GROUP] [NAME | -l label] | TYPE[.VERSION][.GROUP]/NAME ...) [flags] [options]


================================================================================

     kubectl explain --help
List the fields for supported resources.

This command describes the fields associated with each supported API resource.
```

```
Fields are identified via a simple JSONPath identifier:

  <type>.<fieldName>[.<fieldName>]

Add the --recursive flag to display all of the fields at once without
descriptions. Information about each field is retrieved from the server in
OpenAPI format.

Use "kubectl api-resources" for a complete list of supported resources.

Examples:
  # Get the documentation of the resource and its fields
  kubectl explain pods

  # Get the documentation of a specific field of a resource
  kubectl explain pods.spec.containers

Options:
    --api-version='':
       Get different explanations for particular API version (API
       group/version)

    --recursive=false:
       Print the fields of fields (Currently only 1 level deep)

Usage:
  kubectl explain RESOURCE [options]



================================================================================

kubectl edit --help
Edit a resource from the default editor.

 The edit command allows you to directly edit any API resource you can retrieve
via the command-line tools. It will open the editor defined by your KUBE_EDITOR,
or EDITOR environment variables, or fall back to 'vi' for Linux or 'notepad' for
Windows. You can edit multiple objects, although changes are applied one at a
time. The command accepts file names as well as command-line arguments, although
the files you point to must be previously saved versions of resources.

 Editing is done with the API version used to fetch the resource. To edit using
a specific API version, fully-qualify the resource, version, and group.

 The default format is YAML. To edit in JSON, specify "-o json".

 The flag --windows-line-endings can be used to force Windows line endings,
otherwise the default for your operating system will be used.

 In the event an error occurs while updating, a temporary file will be created
on disk that contains your unapplied changes. The most common error when
updating a resource is another editor changing the resource on the server. When
this occurs, you will have to apply your changes to the newer version of the
resource, or update your temporary saved copy to include the latest resource
version.

Examples:
  # Edit the service named 'registry'
  kubectl edit svc/registry

  # Use an alternative editor
  KUBE_EDITOR="nano" kubectl edit svc/registry

  # Edit the job 'myjob' in JSON using the v1 API format
  kubectl edit job.v1.batch/myjob -o json
```

```
  # Edit the deployment 'mydeployment' in YAML and save the modified config in its
annotation
  kubectl edit deployment/mydeployment -o yaml --save-config

  # Edit the deployment/mydeployment's status subresource
  kubectl edit deployment mydeployment --subresource='status'

Options:
    --allow-missing-template-keys=true:
      If true, ignore any errors in templates when a field or map key is
      missing in the template. Only applies to golang and jsonpath output
      formats.

    --field-manager='kubectl-edit':
      Name of the manager used to track field ownership.

    -f, --filename=[ ]:
      Filename, directory, or URL to files to use to edit the resource

    -k, --kustomize='':
      Process the kustomization directory. This flag can't be used together
      with -f or -R.

    -o, --output='':
      Output format. One of: (json, yaml, name, go-template,
      go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
      jsonpath-file).

    --output-patch=false:
      Output the patch if the resource is edited.

    -R, --recursive=false:
      Process the directory used in -f, --filename recursively. Useful when
      you want to manage related manifests organized within the same
      directory.

    --save-config=false:
      If true, the configuration of current object will be saved in its
      annotation. Otherwise, the annotation will be unchanged. This flag is
      useful when you want to perform kubectl apply on this object in the
      future.

    --show-managed-fields=false:
      If true, keep the managedFields when printing objects in JSON or YAML
      format.

    --subresource='':
      If specified, edit will operate on the subresource of the requested
      object. Must be one of [status]. This flag is alpha and may change in
      the future.

    --template='':
      Template string or path to template file to use when -o=go-template,
      -o=go-template-file. The template format is golang templates
      [http://golang.org/pkg/text/template/#pkg-overview].

    --validate='strict':
      Must be one of: strict (or true), warn, ignore (or false).
             "true" or
      "strict" will use a schema to validate the input and fail the request
      if invalid. It will perform server side validation if
      ServerSideFieldValidation is enabled on the api-server, but will fall
      back to less reliable client-side validation if not.
            "warn" will
      warn about unknown or duplicate fields without blocking the request if
```

```
       server-side field validation is enabled on the API server, and behave
       as "ignore" otherwise.
              "false" or "ignore" will not perform any
       schema validation, silently dropping any unknown or duplicate fields.

    --windows-line-endings=false:
       Defaults to the line ending native to your platform.

Usage:
  kubectl edit (RESOURCE/NAME | -f FILENAME) [options]


================================================================================

kubectl delete --help
Delete resources by file names, stdin, resources and names, or by resources and
label selector.

 JSON and YAML formats are accepted. Only one type of argument may be specified:
file names, resources and names, or resources and label selector.

 Some resources, such as pods, support graceful deletion. These resources define
a default period before they are forcibly terminated (the grace period) but you
may override that value with the --grace-period flag, or pass --now to set a
grace-period of 1. Because these resources often represent entities in the
cluster, deletion may not be acknowledged immediately. If the node hosting a pod
is down or cannot reach the API server, termination may take significantly
longer than the grace period. To force delete a resource, you must specify the
--force flag. Note: only a subset of resources support graceful deletion. In
absence of the support, the --grace-period flag is ignored.

 IMPORTANT: Force deleting pods does not wait for confirmation that the pod's
processes have been terminated, which can leave those processes running until
the node detects the deletion and completes graceful deletion. If your processes
use shared storage or talk to a remote API and depend on the name of the pod to
identify themselves, force deleting those pods may result in multiple processes
running on different machines using the same identification which may lead to
data corruption or inconsistency. Only force delete pods when you are sure the
pod is terminated, or if your application can tolerate multiple copies of the
same pod running at once. Also, if you force delete pods, the scheduler may
place new pods on those nodes before the node has released those resources and
causing those pods to be evicted immediately.

 Note that the delete command does NOT do resource version checks, so if someone
submits an update to a resource right when you submit a delete, their update
will be lost along with the rest of the resource.

 After a CustomResourceDefinition is deleted, invalidation of discovery cache
may take up to 6 hours. If you don't want to wait, you might want to run
"kubectl api-resources" to refresh the discovery cache.

Examples:
  # Delete a pod using the type and name specified in pod.json
  kubectl delete -f ./pod.json

  # Delete resources from a directory containing kustomization.yaml - e.g.
dir/kustomization.yaml
  kubectl delete -k dir

  # Delete resources from all files that end with '.json' - i.e. expand wildcard characters
in file names
  kubectl delete -f '*.json'

  # Delete a pod based on the type and name in the JSON passed into stdin
  cat pod.json | kubectl delete -f -
```

```
  # Delete pods and services with same names "baz" and "foo"
  kubectl delete pod,service baz foo

  # Delete pods and services with label name=myLabel
  kubectl delete pods,services -l name=myLabel

  # Delete a pod with minimal delay
  kubectl delete pod foo --now

  # Force delete a pod on a dead node
  kubectl delete pod foo --force

  # Delete all pods
  kubectl delete pods --all

Options:
    --all=false:
       Delete all resources, in the namespace of the specified resource
       types.

    -A, --all-namespaces=false:
       If present, list the requested object(s) across all namespaces.
       Namespace in current context is ignored even if specified with
       --namespace.

    --cascade='background':
       Must be "background", "orphan", or "foreground". Selects the deletion
       cascading strategy for the dependents (e.g. Pods created by a
       ReplicationController). Defaults to background.

    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    --field-selector='':
       Selector (field query) to filter on, supports '=', '==', and
       '!='.(e.g. --field-selector key1=value1,key2=value2). The server only
       supports a limited number of field queries per type.

    -f, --filename=[ ]:
       containing the resource to delete.

    --force=false:
       If true, immediately remove resources from API and bypass graceful
       deletion. Note that immediate deletion of some resources may result in
       inconsistency or data loss and requires confirmation.

    --grace-period=-1:
       Period of time in seconds given to the resource to terminate
       gracefully. Ignored if negative. Set to 1 for immediate shutdown. Can
       only be set to 0 when --force is true (force deletion).

    --ignore-not-found=false:
       Treat "resource not found" as a successful delete. Defaults to "true"
       when --all is specified.

    -k, --kustomize='':
       Process a kustomization directory. This flag can't be used together
       with -f or -R.

    --now=false:
       If true, resources are signaled for immediate shutdown (same as
       --grace-period=1).

    -o, --output='':
```

```
       Output mode. Use "-o name" for shorter output (resource/name).

    --raw='':
       Raw URI to DELETE to the server.  Uses the transport specified by the
       kubeconfig file.

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

    --timeout=0s:
       The length of time to wait before giving up on a delete, zero means
       determine a timeout from the size of the object

    --wait=true:
       If true, wait for resources to be gone before returning. This waits
       for finalizers.

Usage:
  kubectl delete ([-f FILENAME] | [-k DIRECTORY] | TYPE [(NAME | -l label | --all)])
[options]

================================================================================

kubectl rollout --help
Manage the rollout of one or many resources.

 Valid resource types include:

  *  deployments
  *  daemonsets
  *  statefulsets

Examples:
  # Rollback to the previous deployment
  kubectl rollout undo deployment/abc

  # Check the rollout status of a daemonset
  kubectl rollout status daemonset/foo

  # Restart a deployment
  kubectl rollout restart deployment/abc

  # Restart deployments with the app=nginx label
  kubectl rollout restart deployment --selector=app=nginx

Available Commands:
  history       View rollout history
  pause         Mark the provided resource as paused
  restart       Restart a resource
  resume        Resume a paused resource
  status        Show the status of the rollout
  undo          Undo a previous rollout

Usage:
  kubectl rollout SUBCOMMAND [options]


================================================================================
```

```
kubectl scale --help
Set a new size for a deployment, replica set, replication controller, or
stateful set.

 Scale also allows users to specify one or more preconditions for the scale
action.

 If --current-replicas or --resource-version is specified, it is validated
before the scale is attempted, and it is guaranteed that the precondition holds
true when the scale is sent to the server.

Examples:
  # Scale a replica set named 'foo' to 3
  kubectl scale --replicas=3 rs/foo

  # Scale a resource identified by type and name specified in "foo.yaml" to 3
  kubectl scale --replicas=3 -f foo.yaml

  # If the deployment named mysql's current size is 2, scale mysql to 3
  kubectl scale --current-replicas=2 --replicas=3 deployment/mysql

  # Scale multiple replication controllers
  kubectl scale --replicas=5 rc/foo rc/bar rc/baz

  # Scale stateful set named 'web' to 3
  kubectl scale --replicas=3 statefulset/web

Options:
    --all=false:
       Select all resources in the namespace of the specified resource types

    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --current-replicas=-1:
       Precondition for current size. Requires that the current size of the
       resource match this value in order to scale. -1 (default) for no
       condition.

    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    -f, --filename=[ ]:
       Filename, directory, or URL to files identifying the resource to set a
       new size

    -k, --kustomize='':
       Process the kustomization directory. This flag can't be used together
       with -f or -R.

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file).

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    --replicas=0:
       The new desired number of replicas. Required.
```

```
    --resource-version='':
       Precondition for resource version. Requires that the current resource
       version match this value in order to scale.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    --timeout=0s:
       The length of time to wait before giving up on a scale operation, zero
       means don't wait. Any other values should contain a corresponding time
       unit (e.g. 1s, 2m, 3h).

Usage:
  kubectl scale [--resource-version=version] [--current-replicas=count] --replicas=COUNT (-f
FILENAME | TYPE NAME) [options]
```

```
================================================================================
kubectl autoscale --help
Creates an autoscaler that automatically chooses and sets the number of pods that run in a
Kubernetes cluster.

 Looks up a deployment, replica set, stateful set, or replication controller by
name and creates an autoscaler that uses the given resource as a reference. An
autoscaler can automatically increase or decrease number of pods deployed within
the system as needed.

Examples:
  # Auto scale a deployment "foo", with the number of pods between 2 and 10, no target CPU
utilization specified so a default autoscaling policy will be used
  kubectl autoscale deployment foo --min=2 --max=10

  # Auto scale a replication controller "foo", with the number of pods between 1 and 5,
target CPU utilization at 80%
  kubectl autoscale rc foo --max=5 --cpu-percent=80

Options:
    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --cpu-percent=-1:
       The target average CPU utilization (represented as a percent of
       requested CPU) over all the pods. If it's not specified or negative, a
       default autoscaling policy will be used.

    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
```

```
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    --field-manager='kubectl-autoscale':
       Name of the manager used to track field ownership.

    -f, --filename=[ ]:
       Filename, directory, or URL to files identifying the resource to
       autoscale.

    -k, --kustomize='':
       Process the kustomization directory. This flag can't be used together
       with -f or -R.

    --max=-1:
       The upper limit for the number of pods that can be set by the
       autoscaler. Required.

    --min=-1:
       The lower limit for the number of pods that can be set by the
       autoscaler. If it's not specified or negative, the server will apply a
       default value.

    --name='':
       The name for the newly created object. If not specified, the name of
       the input resource will be used.

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file).

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    --save-config=false:
       If true, the configuration of current object will be saved in its
       annotation. Otherwise, the annotation will be unchanged. This flag is
       useful when you want to perform kubectl apply on this object in the
       future.

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

Usage:
  kubectl autoscale (-f FILENAME | TYPE NAME | TYPE/NAME) [--min=MINPODS] --max=MAXPODS [--
cpu-percent=CPU] [options]

================================================================================

kubectl certificate --help
Modify certificate resources.

Available Commands:
  approve       Approve a certificate signing request
  deny          Deny a certificate signing request

Usage:
```

```
  kubectl certificate SUBCOMMAND [options]


================================================================================

kubectl cluster-info --help
Display addresses of the control plane and services with label
kubernetes.io/cluster-service=true. To further debug and diagnose cluster
problems, use 'kubectl cluster-info dump'.

Examples:
  # Print the address of the control plane and cluster services
  kubectl cluster-info

Available Commands:
  dump          Dump relevant information for debugging and diagnosis

Usage:
  kubectl cluster-info [flags] [options]

================================================================================

kubectl top --help
Display Resource (CPU/Memory) usage.

 The top command allows you to see the resource consumption for nodes or pods.

 This command requires Metrics Server to be correctly configured and working on
the server.

Available Commands:
  node          Display resource (CPU/memory) usage of nodes
  pod           Display resource (CPU/memory) usage of pods

Usage:
  kubectl top [flags] [options]


================================================================================

kubectl cordon --help
Mark node as unschedulable.

Examples:
  # Mark node "foo" as unschedulable
  kubectl cordon foo

Options:
    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

Usage:
  kubectl cordon NODE [options]


================================================================================

kubectl uncordon --help
Mark node as schedulable.
```

```
Examples:
  # Mark node "foo" as schedulable
  kubectl uncordon foo

Options:
    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

Usage:
  kubectl uncordon NODE [options]


================================================================================

kubectl drain --help
Drain node in preparation for maintenance.

 The given node will be marked unschedulable to prevent new pods from arriving.
'drain' evicts the pods if the API server supports
https://kubernetes.io/docs/concepts/workloads/pods/disruptions/ eviction
https://kubernetes.io/docs/concepts/workloads/pods/disruptions/ . Otherwise, it
will use normal DELETE to delete the pods. The 'drain' evicts or deletes all
pods except mirror pods (which cannot be deleted through the API server).  If
there are daemon set-managed pods, drain will not proceed without
--ignore-daemonsets, and regardless it will not delete any daemon set-managed
pods, because those pods would be immediately replaced by the daemon set
controller, which ignores unschedulable markings.  If there are any pods that
are neither mirror pods nor managed by a replication controller, replica set,
daemon set, stateful set, or job, then drain will not delete any pods unless you
use --force.  --force will also allow deletion to proceed if the managing
resource of one or more pods is missing.

 'drain' waits for graceful termination. You should not operate on the machine
until the command completes.

 When you are ready to put the node back into service, use kubectl uncordon,
which will make the node schedulable again.

https://kubernetes.io/images/docs/kubectl_drain.svg
Workflowhttps://kubernetes.io/images/docs/kubectl_drain.svg

Examples:
  # Drain node "foo", even if there are pods not managed by a replication controller,
replica set, job, daemon set or stateful set on it
  kubectl drain foo --force

  # As above, but abort if there are pods not managed by a replication controller, replica
set, job, daemon set or stateful set, and use a grace period of 15 minutes
  kubectl drain foo --grace-period=900

Options:
    --chunk-size=500:
       Return large lists in chunks rather than all at once. Pass 0 to
       disable. This flag is beta and may change in the future.

    --delete-emptydir-data=false:
       Continue even if there are pods using emptyDir (local data that will
       be deleted when the node is drained).
```

```
    --disable-eviction=false:
       Force drain to use delete, even if eviction is supported. This will
       bypass checking PodDisruptionBudgets, use with caution.

    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    --force=false:
       Continue even if there are pods that do not declare a controller.

    --grace-period=-1:
       Period of time in seconds given to each pod to terminate gracefully.
       If negative, the default value specified in the pod will be used.

    --ignore-daemonsets=false:
       Ignore DaemonSet-managed pods.

    --pod-selector='':
       Label selector to filter pods on the node

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

    --skip-wait-for-delete-timeout=0:
       If pod DeletionTimestamp older than N seconds, skip waiting for the
       pod.  Seconds must be greater than 0 to skip.

    --timeout=0s:
       The length of time to wait before giving up, zero means infinite

Usage:
  kubectl drain NODE [options]


================================================================================

kubectl taint --help
Update the taints on one or more nodes.

  *  A taint consists of a key, value, and effect. As an argument here, it is
expressed as key=value:effect.
  *  The key must begin with a letter or number, and may contain letters,
numbers, hyphens, dots, and underscores, up to 253 characters.
  *  Optionally, the key can begin with a DNS subdomain prefix and a single '/',
like example.com/my-app.
  *  The value is optional. If given, it must begin with a letter or number, and
may contain letters, numbers, hyphens, dots, and underscores, up to 63
characters.
  *  The effect must be NoSchedule, PreferNoSchedule or NoExecute.
  *  Currently taint can only apply to node.

Examples:
  # Update node 'foo' with a taint with key 'dedicated' and value 'special-user' and effect
'NoSchedule'
  # If a taint with that key and effect already exists, its value is replaced as specified
  kubectl taint nodes foo dedicated=special-user:NoSchedule

  # Remove from node 'foo' the taint with key 'dedicated' and effect 'NoSchedule' if one
exists
  kubectl taint nodes foo dedicated:NoSchedule-
```

```
  # Remove from node 'foo' all the taints with key 'dedicated'
  kubectl taint nodes foo dedicated-

  # Add a taint with key 'dedicated' on nodes having label mylabel=X
  kubectl taint node -l myLabel=X  dedicated=foo:PreferNoSchedule

  # Add to node 'foo' a taint with key 'bar' and no value
  kubectl taint nodes foo bar:NoSchedule

Options:
    --all=false:
       Select all nodes in the cluster

    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    --field-manager='kubectl-taint':
       Name of the manager used to track field ownership.

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file).

    --overwrite=false:
       If true, allow taints to be overwritten, otherwise reject taint
       updates that overwrite existing taints.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    --validate='strict':
       Must be one of: strict (or true), warn, ignore (or false).
              "true" or
       "strict" will use a schema to validate the input and fail the request
       if invalid. It will perform server side validation if
       ServerSideFieldValidation is enabled on the api-server, but will fall
       back to less reliable client-side validation if not.
              "warn" will
       warn about unknown or duplicate fields without blocking the request if
       server-side field validation is enabled on the API server, and behave
       as "ignore" otherwise.
              "false" or "ignore" will not perform any
       schema validation, silently dropping any unknown or duplicate fields.

Usage:
  kubectl taint NODE NAME KEY_1=VAL_1:TAINT_EFFECT_1 ... KEY_N=VAL_N:TAINT_EFFECT_N
[options]
```

```
================================================================================

kubectl describe --help
Show details of a specific resource or group of resources.

 Print a detailed description of the selected resources, including related
resources such as events or controllers. You may select a single object by name,
all objects of that type, provide a name prefix, or label selector. For example:

  $ kubectl describe TYPE NAME_PREFIX

 will first check for an exact match on TYPE and NAME_PREFIX. If no such
resource exists, it will output details for every resource that has a name
prefixed with NAME_PREFIX.

Use "kubectl api-resources" for a complete list of supported resources.

Examples:
  # Describe a node
  kubectl describe nodes kubernetes-node-emt8.c.myproject.internal

  # Describe a pod
  kubectl describe pods/nginx

  # Describe a pod identified by type and name in "pod.json"
  kubectl describe -f pod.json

  # Describe all pods
  kubectl describe pods

  # Describe pods by label name=myLabel
  kubectl describe po -l name=myLabel

  # Describe all pods managed by the 'frontend' replication controller
  # (rc-created pods get the name of the rc as a prefix in the pod name)
  kubectl describe pods frontend

Options:
    -A, --all-namespaces=false:
       If present, list the requested object(s) across all namespaces.
       Namespace in current context is ignored even if specified with
       --namespace.

    --chunk-size=500:
       Return large lists in chunks rather than all at once. Pass 0 to
       disable. This flag is beta and may change in the future.

    -f, --filename=[ ]:
       Filename, directory, or URL to files containing the resource to
       describe

    -k, --kustomize='':
       Process the kustomization directory. This flag can't be used together
       with -f or -R.

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.
```

```
    --show-events=true:
      If true, display events related to the described object.

Usage:
  kubectl describe (-f FILENAME | TYPE [NAME_PREFIX | -l label] | TYPE/NAME)
[options]


================================================================================

kubectl logs --help
Print the logs for a container in a pod or specified resource. If the pod has
only one container, the container name is optional.

Examples:
  # Return snapshot logs from pod nginx with only one container
  kubectl logs nginx

  # Return snapshot logs from pod nginx with multi containers
  kubectl logs nginx --all-containers=true

  # Return snapshot logs from all containers in pods defined by label app=nginx
  kubectl logs -l app=nginx --all-containers=true

  # Return snapshot of previous terminated ruby container logs from pod web-1
  kubectl logs -p -c ruby web-1

  # Begin streaming the logs of the ruby container in pod web-1
  kubectl logs -f -c ruby web-1

  # Begin streaming the logs from all containers in pods defined by label app=nginx
  kubectl logs -f -l app=nginx --all-containers=true

  # Display only the most recent 20 lines of output in pod nginx
  kubectl logs --tail=20 nginx

  # Show all logs from pod nginx written in the last hour
  kubectl logs --since=1h nginx

  # Show logs from a kubelet with an expired serving certificate
  kubectl logs --insecure-skip-tls-verify-backend nginx

  # Return snapshot logs from first container of a job named hello
  kubectl logs job/hello

  # Return snapshot logs from container nginx-1 of a deployment named nginx
  kubectl logs deployment/nginx -c nginx-1

Options:
    --all-containers=false:
      Get all containers' logs in the pod(s).

    -c, --container='':
      Print the logs of this container

    -f, --follow=false:
      Specify if the logs should be streamed.

    --ignore-errors=false:
      If watching / following pod logs, allow for any errors that occur to
      be non-fatal

    --insecure-skip-tls-verify-backend=false:
      Skip verifying the identity of the kubelet that logs are requested
      from.  In theory, an attacker could provide invalid log content back.
```

```
      You might want to use this if your kubelet serving certificates have
      expired.

    --limit-bytes=0:
      Maximum bytes of logs to return. Defaults to no limit.

    --max-log-requests=5:
      Specify maximum number of concurrent logs to follow when using by a
      selector. Defaults to 5.

    --pod-running-timeout=20s:
      The length of time (like 5s, 2m, or 3h, higher than zero) to wait
      until at least one pod is running

    --prefix=false:
      Prefix each log line with the log source (pod name and container name)

    -p, --previous=false:
      If true, print the logs for the previous instance of the container in
      a pod if it exists.

    -l, --selector='':
      Selector (label query) to filter on, supports '=', '==', and
      '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
      all of the specified label constraints.

    --since=0s:
      Only return logs newer than a relative duration like 5s, 2m, or 3h.
      Defaults to all logs. Only one of since-time / since may be used.

    --since-time='':
      Only return logs after a specific date (RFC3339). Defaults to all
      logs. Only one of since-time / since may be used.

    --tail=-1:
      Lines of recent log file to display. Defaults to -1 with no selector,
      showing all log lines otherwise 10, if a selector is provided.

    --timestamps=false:
      Include timestamps on each line in the log output

Usage:
  kubectl logs [-f] [-p] (POD | TYPE/NAME) [-c CONTAINER] [options]


================================================================================

kubectl attach --help
Attach to a process that is already running inside an existing container.

Examples:
  # Get output from running pod mypod; use the 'kubectl.kubernetes.io/default-container'
annotation
  # for selecting the container to be attached or the first container in the pod will be
chosen
  kubectl attach mypod

  # Get output from ruby-container from pod mypod
  kubectl attach mypod -c ruby-container

  # Switch to raw terminal mode; sends stdin to 'bash' in ruby-container from pod mypod
  # and sends stdout/stderr from 'bash' back to the client
  kubectl attach mypod -c ruby-container -i -t

  # Get output from the first pod of a replica set named nginx
  kubectl attach rs/nginx
```

```
Options:
    -c, --container='':
       Container name. If omitted, use the
       kubectl.kubernetes.io/default-container annotation for selecting the
       container to be attached or the first container in the pod will be
       chosen

    --pod-running-timeout=1m0s:
       The length of time (like 5s, 2m, or 3h, higher than zero) to wait
       until at least one pod is running

    -q, --quiet=false:
       Only print output from the remote session

    -i, --stdin=false:
       Pass stdin to the container

    -t, --tty=false:
       Stdin is a TTY

Usage:
  kubectl attach (POD | TYPE/NAME) -c CONTAINER [options]


================================================================================

kubectl exec --help
Execute a command in a container.

Examples:
  # Get output from running the 'date' command from pod mypod, using the first-container by
default
  kubectl exec mypod -- date

  # Get output from running the 'date' command in ruby-container from pod mypod
  kubectl exec mypod -c ruby-container -- date

  # Switch to raw terminal mode; sends stdin to 'bash' in ruby-container from pod mypod
  # and sends stdout/stderr from 'bash' back to the client
  kubectl exec mypod -c ruby-container -i -t -- bash -il

  # List contents of /usr from the first container of pod mypod and sort by modification
time
  # If the command you want to execute in the pod has any flags in common (e.g. -i),
  # you must use two dashes (--) to separate your command's flags/arguments
  # Also note, do not surround your command and its flags/arguments with quotes
  # unless that is how you would execute it normally (i.e., do ls -t /usr, not "ls -t /usr")
  kubectl exec mypod -i -t -- ls -t /usr

  # Get output from running 'date' command from the first pod of the deployment
mydeployment, using the first container by default
  kubectl exec deploy/mydeployment -- date

  # Get output from running 'date' command from the first pod of the service myservice,
using the first container by default
  kubectl exec svc/myservice -- date

Options:
    -c, --container='':
       Container name. If omitted, use the
       kubectl.kubernetes.io/default-container annotation for selecting the
       container to be attached or the first container in the pod will be
       chosen

    -f, --filename=[ ]:
```

```
       to use to exec into the resource

    --pod-running-timeout=1m0s:
       The length of time (like 5s, 2m, or 3h, higher than zero) to wait
       until at least one pod is running

    -q, --quiet=false:
       Only print output from the remote session

    -i, --stdin=false:
       Pass stdin to the container

    -t, --tty=false:
       Stdin is a TTY

Usage:
  kubectl exec (POD | TYPE/NAME) [-c CONTAINER] [flags] -- COMMAND [args...]
[options]

================================================================================
kubectl port-forward --help
Forward one or more local ports to a pod.

 Use resource type/name such as deployment/mydeployment to select a pod.
Resource type defaults to 'pod' if omitted.

 If there are multiple pods matching the criteria, a pod will be selected
automatically. The forwarding session ends when the selected pod terminates, and
a rerun of the command is needed to resume forwarding.

Examples:
  # Listen on ports 5000 and 6000 locally, forwarding data to/from ports 5000 and 6000 in
the pod
  kubectl port-forward pod/mypod 5000 6000

  # Listen on ports 5000 and 6000 locally, forwarding data to/from ports 5000 and 6000 in a
pod selected by the deployment
  kubectl port-forward deployment/mydeployment 5000 6000

  # Listen on port 8443 locally, forwarding to the targetPort of the service's port named
"https" in a pod selected by the service
  kubectl port-forward service/myservice 8443:https

  # Listen on port 8888 locally, forwarding to 5000 in the pod
  kubectl port-forward pod/mypod 8888:5000

  # Listen on port 8888 on all addresses, forwarding to 5000 in the pod
  kubectl port-forward --address 0.0.0.0 pod/mypod 8888:5000

  # Listen on port 8888 on localhost and selected IP, forwarding to 5000 in the pod
  kubectl port-forward --address localhost,10.19.21.23 pod/mypod 8888:5000

  # Listen on a random port locally, forwarding to 5000 in the pod
  kubectl port-forward pod/mypod :5000

Options:
    --address=[localhost]:
       Addresses to listen on (comma separated). Only accepts IP addresses or
       localhost as a value. When localhost is supplied, kubectl will try to
       bind on both 127.0.0.1 and ::1 and will fail if neither of these
       addresses are available to bind.

    --pod-running-timeout=1m0s:
       The length of time (like 5s, 2m, or 3h, higher than zero) to wait
       until at least one pod is running
```

```
Usage:
  kubectl port-forward TYPE/NAME [options] [LOCAL_PORT:]REMOTE_PORT [...
[LOCAL_PORT_N:]REMOTE_PORT_N]


================================================================================
kubectl proxy --help

Creates a proxy server or application-level gateway between localhost and the
Kubernetes API server. It also allows serving static content over specified HTTP
path. All incoming data enters through one port and gets forwarded to the remote
Kubernetes API server port, except for the path matching the static content
path.

Examples:
  # To proxy all of the Kubernetes API and nothing else
  kubectl proxy --api-prefix=/

  # To proxy only part of the Kubernetes API and also some static files
  # You can get pods info with 'curl localhost:8001/api/v1/pods'
  kubectl proxy --www=/my/files --www-prefix=/static/ --api-prefix=/api/

  # To proxy the entire Kubernetes API at a different root
  # You can get pods info with 'curl localhost:8001/custom/api/v1/pods'
  kubectl proxy --api-prefix=/custom/

  # Run a proxy to the Kubernetes API server on port 8011, serving static content from
./local/www/
  kubectl proxy --port=8011 --www=./local/www/

  # Run a proxy to the Kubernetes API server on an arbitrary local port
  # The chosen port for the server will be output to stdout
  kubectl proxy --port=0

  # Run a proxy to the Kubernetes API server, changing the API prefix to k8s-api
  # This makes e.g. the pods API available at localhost:8001/k8s-api/v1/pods/
  kubectl proxy --api-prefix=/k8s-api

Options:
    --accept-hosts='^localhost$,^127\.0\.0\.1$,^\[::1\]$':
       Regular expression for hosts that the proxy should accept.

    --accept-paths='^.*':
       Regular expression for paths that the proxy should accept.

    --address='127.0.0.1':
       The IP address on which to serve on.

    --api-prefix='/':
       Prefix to serve the proxied API under.

    --append-server-path=false:
       If true, enables automatic path appending of the kube context server
       path to each request.

    --disable-filter=false:
       If true, disable request filtering in the proxy. This is dangerous,
       and can leave you vulnerable to XSRF attacks, when used with an
       accessible port.

    --keepalive=0s:
       keepalive specifies the keep-alive period for an active network
       connection. Set to 0 to disable keepalive.

    -p, --port=8001:
       The port on which to run the proxy. Set to 0 to pick a random port.
```

```
    --reject-methods='^$':
       Regular expression for HTTP methods that the proxy should reject
       (example --reject-methods='POST,PUT,PATCH').

    --reject-paths='^/api/.*/pods/.*/exec,^/api/.*/pods/.*/attach':
       Regular expression for paths that the proxy should reject. Paths
       specified here will be rejected even accepted by --accept-paths.

    -u, --unix-socket='':
       Unix socket on which to run the proxy.

    -w, --www='':
       Also serve static files from the given directory under the specified
       prefix.

    -P, --www-prefix='/static/':
       Prefix to serve static files under, if static file directory is
       specified.

Usage:
  kubectl proxy [--port=PORT] [--www=static-dir] [--www-prefix=prefix] [--api-prefix=prefix]
[options]


================================================================================
kubectl cp --help
Copy files and directories to and from containers.

Examples:
  # !!!Important Note!!!
  # Requires that the 'tar' binary is present in your container
  # image.  If 'tar' is not present, 'kubectl cp' will fail.
  #
  # For advanced use cases, such as symlinks, wildcard expansion or
  # file mode preservation, consider using 'kubectl exec'.

  # Copy /tmp/foo local file to /tmp/bar in a remote pod in namespace
<some-namespace>
  tar cf - /tmp/foo | kubectl exec -i -n <some-namespace> <some-pod> -- tar xf - -C /tmp/bar

  # Copy /tmp/foo from a remote pod to /tmp/bar locally
  kubectl exec -n <some-namespace> <some-pod> -- tar cf - /tmp/foo | tar xf - -C /tmp/bar

  # Copy /tmp/foo_dir local directory to /tmp/bar_dir in a remote pod in the default
namespace
  kubectl cp /tmp/foo_dir <some-pod>:/tmp/bar_dir

  # Copy /tmp/foo local file to /tmp/bar in a remote pod in a specific container
  kubectl cp /tmp/foo <some-pod>:/tmp/bar -c <specific-container>

  # Copy /tmp/foo local file to /tmp/bar in a remote pod in namespace <some-namespace>
  kubectl cp /tmp/foo <some-namespace>/<some-pod>:/tmp/bar

  # Copy /tmp/foo from a remote pod to /tmp/bar locally
  kubectl cp <some-namespace>/<some-pod>:/tmp/foo /tmp/bar

Options:
    -c, --container='':
       Container name. If omitted, use the
       kubectl.kubernetes.io/default-container annotation for selecting the
       container to be attached or the first container in the pod will be
       chosen

    --no-preserve=false:
       The copied file/directory's ownership and permissions will not be
```

```
      preserved in the container

    --retries=0:
      Set number of retries to complete a copy operation from a container.
      Specify 0 to disable or any negative value for infinite retrying. The
      default is 0 (no retry).

Usage:
  kubectl cp <file-spec-src> <file-spec-dest> [options]


================================================================================

kubectl auth --help
Inspect authorization

Available Commands:
  can-i         Check whether an action is allowed
  reconcile     Reconciles rules for RBAC role, role binding, cluster role, and
cluster role binding objects

Usage:
  kubectl auth [flags] [options]


================================================================================

kubectl debug --help
Debug cluster resources using interactive debugging containers.

 'debug' provides automation for common debugging tasks for cluster objects
identified by resource and name. Pods will be used by default if no resource is
specified.

 The action taken by 'debug' varies depending on what resource is specified.
Supported actions include:

  *  Workload: Create a copy of an existing pod with certain attributes changed,
for example changing the image tag to a new version.
  *  Workload: Add an ephemeral container to an already running pod, for example
to add debugging utilities without restarting the pod.
  *  Node: Create a new pod that runs in the node's host namespaces and can
access the node's filesystem.

Examples:
  # Create an interactive debugging session in pod mypod and immediately attach to it.
  # (requires the EphemeralContainers feature to be enabled in the cluster)
  kubectl debug mypod -it --image=busybox

  # Create a debug container named debugger using a custom automated debugging image.
  # (requires the EphemeralContainers feature to be enabled in the cluster)
  kubectl debug --image=myproj/debug-tools -c debugger mypod

  # Create a copy of mypod adding a debug container and attach to it
  kubectl debug mypod -it --image=busybox --copy-to=my-debugger

  # Create a copy of mypod changing the command of mycontainer
  kubectl debug mypod -it --copy-to=my-debugger --container=mycontainer -- sh

  # Create a copy of mypod changing all container images to busybox
  kubectl debug mypod --copy-to=my-debugger --set-image=*=busybox

  # Create a copy of mypod adding a debug container and changing container images
  kubectl debug mypod -it --copy-to=my-debugger --image=debian --set-
image=app=app:debug,sidecar=sidecar:debug
```

```
  # Create an interactive debugging session on a node and immediately attach to it.
  # The container will run in the host namespaces and the host's filesystem will be mounted
at /host
  kubectl debug node/mynode -it --image=busybox

Options:
    --arguments-only=false:
       If specified, everything after -- will be passed to the new container
       as Args instead of Command.

    --attach=false:
       If true, wait for the container to start running, and then attach as
       if 'kubectl attach ...' were called.  Default false, unless
       '-i/--stdin' is set, in which case the default is true.

    -c, --container='':
       Container name to use for debug container.

    --copy-to='':
       Create a copy of the target Pod with this name.

    --env=[ ]:
       Environment variables to set in the container.

    --image='':
       Container image to use for debug container.

    --image-pull-policy='':
       The image pull policy for the container. If left empty, this value
       will not be specified by the client and defaulted by the server.

    --profile='legacy':
       Debugging profile.

    -q, --quiet=false:
       If true, suppress informational messages.

    --replace=false:
       When used with '--copy-to', delete the original Pod.

    --same-node=false:
       When used with '--copy-to', schedule the copy of target Pod on the
       same node.

    --set-image=[ ]:
       When used with '--copy-to', a list of name=image pairs for changing
       container images, similar to how 'kubectl set image' works.

    --share-processes=true:
       When used with '--copy-to', enable process namespace sharing in the
       copy.

    -i, --stdin=false:
       Keep stdin open on the container(s) in the pod, even if nothing is
       attached.

    --target='':
       When using an ephemeral container, target processes in this container
       name.

    -t, --tty=false:
       Allocate a TTY for the debugging container.

Usage:
  kubectl debug (POD | TYPE[[.VERSION].GROUP]/NAME) [ -- COMMAND [args...] ] [options]
```

```
================================================================================

kubectl events --help
Display events

 Prints a table of the most important information about events. You can request
events for a namespace, for all namespace, or filtered to only those pertaining
to a specified resource.

Examples:
  # List recent events in the default namespace.
  kubectl events

  # List recent events in all namespaces.
  kubectl events --all-namespaces

  # List recent events for the specified pod, then wait for more events and list them as
they arrive.
  kubectl events --for pod/web-pod-13je7 --watch

  # List recent events in given format. Supported ones, apart from default, are json and
yaml.
  kubectl events -oyaml

  # List recent only events in given event types
  kubectl events --types=Warning,Normal

Options:
    -A, --all-namespaces=false:
       If present, list the requested object(s) across all namespaces.
       Namespace in current context is ignored even if specified with
       --namespace.

    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --chunk-size=500:
       Return large lists in chunks rather than all at once. Pass 0 to
       disable. This flag is beta and may change in the future.

    --for='':
       Filter events to only those pertaining to the specified resource.

    --no-headers=false:
       When using the default output format, don't print headers.

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file).

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    --types=[ ]:
       Output only events of given types.

    -w, --watch=false:
```

```
       After listing the requested events, watch for more events.

Usage:
  kubectl events
[(-o|--output=)json|yaml|name|go-template|go-template-file|template|templatefile|jsonpath|
jsonpath-as-json|jsonpath-file]
[--for TYPE/NAME] [--watch] [--event=Normal,Warning] [options]


================================================================================

kubectl diff --help

Diff configurations specified by file name or stdin between the current online
configuration, and the configuration as it would be if applied.

 The output is always YAML.

 KUBECTL_EXTERNAL_DIFF environment variable can be used to select your own diff
command. Users can use external commands with params too, example:
KUBECTL_EXTERNAL_DIFF="colordiff -N -u"

 By default, the "diff" command available in your path will be run with the "-u"
(unified diff) and "-N" (treat absent files as empty) options.

 Exit status: 0 No differences were found. 1 Differences were found. >1 Kubectl
or diff failed with an error.

 Note: KUBECTL_EXTERNAL_DIFF, if used, is expected to follow that convention.

Examples:
  # Diff resources included in pod.json
  kubectl diff -f pod.json

  # Diff file read from stdin
  cat service.yaml | kubectl diff -f -

Options:
    --field-manager='kubectl-client-side-apply':
       Name of the manager used to track field ownership.

    -f, --filename=[ ]:
       Filename, directory, or URL to files contains the configuration to
       diff

    --force-conflicts=false:
       If true, server-side apply will force the changes against conflicts.

    -k, --kustomize='':
       Process the kustomization directory. This flag can't be used together
       with -f or -R.

    --prune=false:
       Include resources that would be deleted by pruning. Can be used with
       -l and default shows all resources would be pruned

    --prune-allowlist=[ ]:
       Overwrite the default whitelist with <group/version/kind> for --prune

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
```

```
      '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
      all of the specified label constraints.

    --server-side=false:
      If true, apply runs in the server instead of the client.

    --show-managed-fields=false:
      If true, include managed fields in the diff.

Usage:
  kubectl diff -f FILENAME [options]


================================================================================

kubectl apply --help
Apply a configuration to a resource by file name or stdin. The resource name
must be specified. This resource will be created if it doesn't exist yet. To use
'apply', always create the resource initially with either 'apply' or 'create
--save-config'.

 JSON and YAML formats are accepted.

 Alpha Disclaimer: the --prune functionality is not yet complete. Do not use
unless you are aware of what the current state is. See
https://issues.k8s.io/34274.

Examples:
  # Apply the configuration in pod.json to a pod
  kubectl apply -f ./pod.json

  # Apply resources from a directory containing kustomization.yaml - e.g.
dir/kustomization.yaml
  kubectl apply -k dir/

  # Apply the JSON passed into stdin to a pod
  cat pod.json | kubectl apply -f -

  # Apply the configuration from all files that end with '.json' - i.e. expand wildcard
characters in file names
  kubectl apply -f '*.json'

  # Note: --prune is still in Alpha
  # Apply the configuration in manifest.yaml that matches label app=nginx and delete all
other resources that are not in the file and match label app=nginx
  kubectl apply --prune -f manifest.yaml -l app=nginx

  # Apply the configuration in manifest.yaml and delete all the other config maps that are
not in the file
  kubectl apply --prune -f manifest.yaml --all --prune-allowlist=core/v1/ConfigMap

Available Commands:
  edit-last-applied   Edit latest last-applied-configuration annotations of are
source/object
  set-last-applied    Set the last-applied-configuration annotation on a live object to
match the contents of a file
  view-last-applied   View the latest last-applied-configuration annotations of a
resource/object

Options:
    --all=false:
      Select all resources in the namespace of the specified resource types.

    --allow-missing-template-keys=true:
      If true, ignore any errors in templates when a field or map key is
      missing in the template. Only applies to golang and jsonpath output
```

```
   formats.

--cascade='background':
   Must be "background", "orphan", or "foreground". Selects the deletion
   cascading strategy for the dependents (e.g. Pods created by a
   ReplicationController). Defaults to background.

--dry-run='none':
   Must be "none", "server", or "client". If client strategy, only print
   the object that would be sent, without sending it. If server strategy,
   submit server-side request without persisting the resource.

--field-manager='kubectl-client-side-apply':
   Name of the manager used to track field ownership.

-f, --filename=[ ]:
   The files that contain the configurations to apply.

--force=false:
   If true, immediately remove resources from API and bypass graceful
   deletion. Note that immediate deletion of some resources may result in
   inconsistency or data loss and requires confirmation.

--force-conflicts=false:
   If true, server-side apply will force the changes against conflicts.

--grace-period=-1:
   Period of time in seconds given to the resource to terminate
   gracefully. Ignored if negative. Set to 1 for immediate shutdown. Can
   only be set to 0 when --force is true (force deletion).

-k, --kustomize='':
   Process a kustomization directory. This flag can't be used together
   with -f or -R.

--openapi-patch=true:
   If true, use openapi to calculate diff when the openapi presents and
   the resource can be found in the openapi spec. Otherwise, fall back to
   use baked-in types.

-o, --output='':
   Output format. One of: (json, yaml, name, go-template,
   go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
   jsonpath-file).

--overwrite=true:
   Automatically resolve conflicts between the modified and live
   configuration by using values from the modified configuration

--prune=false:
   Automatically delete resource objects, that do not appear in the
   configs and are created by either apply or create --save-config.
   Should be used with either -l or --all.

--prune-allowlist=[ ]:
   Overwrite the default allowlist with <group/version/kind> for --prune

-R, --recursive=false:
   Process the directory used in -f, --filename recursively. Useful when
   you want to manage related manifests organized within the same
   directory.

-l, --selector='':
   Selector (label query) to filter on, supports '=', '==', and
   '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
   all of the specified label constraints.
```

```
    --server-side=false:
       If true, apply runs in the server instead of the client.

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    --timeout=0s:
       The length of time to wait before giving up on a delete, zero means
       determine a timeout from the size of the object

    --validate='strict':
       Must be one of: strict (or true), warn, ignore (or false).
             "true" or
       "strict" will use a schema to validate the input and fail the request
       if invalid. It will perform server side validation if
       ServerSideFieldValidation is enabled on the api-server, but will fall
       back to less reliable client-side validation if not.
             "warn" will
       warn about unknown or duplicate fields without blocking the request if
       server-side field validation is enabled on the API server, and behave
       as "ignore" otherwise.
             "false" or "ignore" will not perform any
       schema validation, silently dropping any unknown or duplicate fields.

    --wait=false:
       If true, wait for resources to be gone before returning. This waits
       for finalizers.

Usage:
  kubectl apply (-f FILENAME | -k DIRECTORY) [options]

================================================================================

kubectl patch --help
Update fields of a resource using strategic merge patch, a JSON merge patch, or
a JSON patch.

 JSON and YAML formats are accepted.

 Note: Strategic merge patch is not supported for custom resources.

Examples:
  # Partially update a node using a strategic merge patch, specifying the patch as JSON
  kubectl patch node k8s-node-1 -p '{"spec":{"unschedulable":true}}'

  # Partially update a node using a strategic merge patch, specifying the patch as YAML
  kubectl patch node k8s-node-1 -p $'spec:\n unschedulable: true'

  # Partially update a node identified by the type and name specified in "node.json" using
strategic merge patch
  kubectl patch -f node.json -p '{"spec":{"unschedulable":true}}'

  # Update a container's image; spec.containers[*].name is required because it's a merge key
  kubectl patch pod valid-pod -p
'{"spec":{"containers":[{"name":"kubernetes-serve-hostname","image":"newimage"}]}}'

  # Update a container's image using a JSON patch with positional arrays
  kubectl patch pod valid-pod --type='json' -p='[{"op": "replace", "path":
"/spec/containers/0/image", "value":"new image"}]'
```

```
  # Update a deployment's replicas through the scale subresource using a merge patch.
  kubectl patch deployment nginx-deployment --subresource='scale' --type='merge' -p
'{"spec":{"replicas":2}}'

Options:
    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    --field-manager='kubectl-patch':
       Name of the manager used to track field ownership.

    -f, --filename=[ ]:
       Filename, directory, or URL to files identifying the resource to
       update

    -k, --kustomize='':
       Process the kustomization directory. This flag can't be used together
       with -f or -R.

    --local=false:
       If true, patch will operate on the content of the file, not the
       server-side resource.

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file).

    -p, --patch='':
       The patch to be applied to the resource JSON file.

    --patch-file='':
       A file containing a patch to be applied to the resource.

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --subresource='':
       If specified, patch will operate on the subresource of the requested
       object. Must be one of [status scale]. This flag is alpha and may
       change in the future.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    --type='strategic':
       The type of patch being provided; one of [json merge strategic]

Usage:
  kubectl patch (-f FILENAME | TYPE NAME) [-p PATCH|--patch-file FILE] [options]
```

```
================================================================================

kubectl replace --help
Replace a resource by file name or stdin.

 JSON and YAML formats are accepted. If replacing an existing resource, the
complete resource spec must be provided. This can be obtained by

  $ kubectl get TYPE NAME -o yaml

Examples:
  # Replace a pod using the data in pod.json
  kubectl replace -f ./pod.json

  # Replace a pod based on the JSON passed into stdin
  cat pod.json | kubectl replace -f -

  # Update a single-container pod's image version (tag) to v4
  kubectl get pod mypod -o yaml | sed 's/\(image: myimage\):.*$/\1:v4/' | kubectl replace -f
-

  # Force replace, delete and then re-create the resource
  kubectl replace --force -f ./pod.json

Options:
    --allow-missing-template-keys=true:
      If true, ignore any errors in templates when a field or map key is
      missing in the template. Only applies to golang and jsonpath output
      formats.

    --cascade='background':
      Must be "background", "orphan", or "foreground". Selects the deletion
      cascading strategy for the dependents (e.g. Pods created by a
      ReplicationController). Defaults to background.

    --dry-run='none':
      Must be "none", "server", or "client". If client strategy, only print
      the object that would be sent, without sending it. If server strategy,
      submit server-side request without persisting the resource.

    --field-manager='kubectl-replace':
      Name of the manager used to track field ownership.

    -f, --filename=[ ]:
      The files that contain the configurations to replace.

    --force=false:
      If true, immediately remove resources from API and bypass graceful
      deletion. Note that immediate deletion of some resources may result in
      inconsistency or data loss and requires confirmation.

    --grace-period=-1:
      Period of time in seconds given to the resource to terminate
      gracefully. Ignored if negative. Set to 1 for immediate shutdown. Can
      only be set to 0 when --force is true (force deletion).

    -k, --kustomize='':
      Process a kustomization directory. This flag can't be used together
      with -f or -R.

    -o, --output='':
      Output format. One of: (json, yaml, name, go-template,
      go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
      jsonpath-file).
```

```
    --raw='':
       Raw URI to PUT to the server.  Uses the transport specified by the
       kubeconfig file.

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    --save-config=false:
       If true, the configuration of current object will be saved in its
       annotation. Otherwise, the annotation will be unchanged. This flag is
       useful when you want to perform kubectl apply on this object in the
       future.

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --subresource='':
       If specified, replace will operate on the subresource of the requested
       object. Must be one of [status scale]. This flag is alpha and may
       change in the future.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    --timeout=0s:
       The length of time to wait before giving up on a delete, zero means
       determine a timeout from the size of the object

    --validate='strict':
       Must be one of: strict (or true), warn, ignore (or false).
              "true" or
       "strict" will use a schema to validate the input and fail the request
       if invalid. It will perform server side validation if
       ServerSideFieldValidation is enabled on the api-server, but will fall
       back to less reliable client-side validation if not.
              "warn" will
       warn about unknown or duplicate fields without blocking the request if
       server-side field validation is enabled on the API server, and behave
       as "ignore" otherwise.
              "false" or "ignore" will not perform any
       schema validation, silently dropping any unknown or duplicate fields.

    --wait=false:
       If true, wait for resources to be gone before returning. This waits
       for finalizers.

Usage:
  kubectl replace -f FILENAME [options]


================================================================================

kubectl wait --help
Experimental: Wait for a specific condition on one or many resources.

 The command takes multiple resources and waits until the specified condition is
seen in the Status field of every given resource.

 Alternatively, the command can wait for the given set of resources to be
deleted by providing the "delete" keyword as the value to the --for flag.
```

```
 A successful message will be printed to stdout indicating when the specified
condition has been met. You can use -o option to change to output destination.

Examples:
  # Wait for the pod "busybox1" to contain the status condition of type "Ready"
  kubectl wait --for=condition=Ready pod/busybox1

  # The default value of status condition is true; you can wait for other targets after an
equal delimiter (compared after Unicode simple case folding, which is a more general form of
case-insensitivity):
  kubectl wait --for=condition=Ready=false pod/busybox1

  # Wait for the pod "busybox1" to contain the status phase to be "Running".
  kubectl wait --for=jsonpath='{.status.phase}'=Running pod/busybox1

  # Wait for the pod "busybox1" to be deleted, with a timeout of 60s, after having issued
the "delete" command
  kubectl delete pod/busybox1
  kubectl wait --for=delete pod/busybox1 --timeout=60s

Options:
    --all=false:
       Select all resources in the namespace of the specified resource types

    -A, --all-namespaces=false:
       If present, list the requested object(s) across all namespaces.
       Namespace in current context is ignored even if specified with
       --namespace.

    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --field-selector='':
       Selector (field query) to filter on, supports '=', '==', and
       '!='.(e.g. --field-selector key1=value1,key2=value2). The server only
       supports a limited number of field queries per type.

    -f, --filename=[ ]:
       identifying the resource.

    --for='':
       The condition to wait on:
       [delete|condition=condition-name[=condition-value]|jsonpath='{JSONPath
       expression}'=JSONPath Condition]. The default condition-value is true.
       Condition values are compared after Unicode simple case folding, which
       is a more general form of case-insensitivity.

    --local=false:
       If true, annotation will NOT contact api-server but run locally.

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file).

    -R, --recursive=true:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2)
```

```
    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

    --timeout=30s:
       The length of time to wait before giving up.  Zero means check once
       and don't wait, negative means wait for a week.

Usage:
  kubectl wait ([-f FILENAME] | resource.group/resource.name | resource.group [(-l label |
--all)]) [--for=delete|--for condition=available|--for=jsonpath='{}'=value] [options]


================================================================================

kubectl kustomize --help
Build a set of KRM resources using a 'kustomization.yaml' file. The DIR argument
must be a path to a directory containing 'kustomization.yaml', or a git
repository URL with a path suffix specifying same with respect to the repository
root. If DIR is omitted, '.' is assumed.

Examples:
  # Build the current working directory
  kubectl kustomize

  # Build some shared configuration directory
  kubectl kustomize /home/config/production

  # Build from github
  kubectl kustomize
https://github.com/kubernetes-sigs/kustomize.git/examples/helloWorld?ref=v1.0.6

Options:
    --as-current-user=false:
       use the uid and gid of the command executor to run the function in the
       container

    --enable-alpha-plugins=false:
       enable kustomize plugins

    --enable-helm=false:
       Enable use of the Helm chart inflator generator.

    -e, --env=[ ]:
       a list of environment variables to be used by functions

    --helm-command='helm':
       helm command (path to executable)

    --load-restrictor='LoadRestrictionsRootOnly':
       if set to 'LoadRestrictionsNone', local kustomizations may load files
       from outside their root. This does, however, break the relocatability
       of the kustomization.

    --mount=[ ]:
       a list of storage options read from the filesystem

    --network=false:
       enable network access for functions that declare it
```

```
    --network-name='bridge':
       the docker network to run the container in

    -o, --output='':
       If specified, write output to this path.

    --reorder='legacy':
       Reorder the resources just before output. Use 'legacy' to apply a
       legacy reordering (Namespaces first, Webhooks last, etc). Use 'none'
       to suppress a final reordering.

Usage:
  kubectl kustomize DIR [flags] [options]


================================================================================

kubectl label --help
Update the labels on a resource.

  *  A label key and value must begin with a letter or number, and may contain
letters, numbers, hyphens, dots, and underscores, up to 63 characters each.
  *  Optionally, the key can begin with a DNS subdomain prefix and a single '/',
like example.com/my-app.
  *  If --overwrite is true, then existing labels can be overwritten, otherwise
attempting to overwrite a label will result in an error.
  *  If --resource-version is specified, then updates will use this resource
version, otherwise the existing resource-version will be used.

Examples:
  # Update pod 'foo' with the label 'unhealthy' and the value 'true'
  kubectl label pods foo unhealthy=true

  # Update pod 'foo' with the label 'status' and the value 'unhealthy', overwriting any
existing value
  kubectl label --overwrite pods foo status=unhealthy

  # Update all pods in the namespace
  kubectl label pods --all status=unhealthy

  # Update a pod identified by the type and name in "pod.json"
  kubectl label -f pod.json status=unhealthy

  # Update pod 'foo' only if the resource is unchanged from version 1
  kubectl label pods foo status=unhealthy --resource-version=1

  # Update pod 'foo' by removing a label named 'bar' if it exists
  # Does not require the --overwrite flag
  kubectl label pods foo bar-

Options:
    --all=false:
       Select all resources, in the namespace of the specified resource types

    -A, --all-namespaces=false:
       If true, check the specified action in all namespaces.

    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.
```

```
    --field-manager='kubectl-label':
       Name of the manager used to track field ownership.

    --field-selector='':
       Selector (field query) to filter on, supports '=', '==', and
       '!='.(e.g. --field-selector key1=value1,key2=value2). The server only
       supports a limited number of field queries per type.

    -f, --filename=[ ]:
       Filename, directory, or URL to files identifying the resource to
       update the labels

    -k, --kustomize='':
       Process the kustomization directory. This flag can't be used together
       with -f or -R.

    --list=false:
       If true, display the labels for a given resource.

    --local=false:
       If true, label will NOT contact api-server but run locally.

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file).

    --overwrite=false:
       If true, allow labels to be overwritten, otherwise reject label
       updates that overwrite existing labels.

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    --resource-version='':
       If non-empty, the labels update will only succeed if this is the
       current resource-version for the object. Only valid when specifying a
       single resource.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

Usage:
  kubectl label [--overwrite] (-f FILENAME | TYPE NAME) KEY_1=VAL_1 ... KEY_N=VAL_N [--
resource-version=version] [options]


================================================================================

kubectl annotate --help
Update the annotations on one or more resources.
```

```
 All Kubernetes objects support the ability to store additional data with the
object as annotations. Annotations are key/value pairs that can be larger than
labels and include arbitrary string values such as structured JSON. Tools and
system extensions may use annotations to store their own data.

 Attempting to set an annotation that already exists will fail unless
--overwrite is set. If --resource-version is specified and does not match the
current resource version on the server the command will fail.

Use "kubectl api-resources" for a complete list of supported resources.

Examples:
  # Update pod 'foo' with the annotation 'description' and the value 'my frontend'
  # If the same annotation is set multiple times, only the last value will be applied
  kubectl annotate pods foo description='my frontend'

  # Update a pod identified by type and name in "pod.json"
  kubectl annotate -f pod.json description='my frontend'

  # Update pod 'foo' with the annotation 'description' and the value 'my frontend running
nginx', overwriting any existing value
  kubectl annotate --overwrite pods foo description='my frontend running nginx'

  # Update all pods in the namespace
  kubectl annotate pods --all description='my frontend running nginx'

  # Update pod 'foo' only if the resource is unchanged from version 1
  kubectl annotate pods foo description='my frontend running nginx'
--resource-version=1

  # Update pod 'foo' by removing an annotation named 'description' if it exists
  # Does not require the --overwrite flag
  kubectl annotate pods foo description-

Options:
    --all=false:
       Select all resources, in the namespace of the specified resource
       types.

    -A, --all-namespaces=false:
       If true, check the specified action in all namespaces.

    --allow-missing-template-keys=true:
       If true, ignore any errors in templates when a field or map key is
       missing in the template. Only applies to golang and jsonpath output
       formats.

    --dry-run='none':
       Must be "none", "server", or "client". If client strategy, only print
       the object that would be sent, without sending it. If server strategy,
       submit server-side request without persisting the resource.

    --field-manager='kubectl-annotate':
       Name of the manager used to track field ownership.

    --field-selector='':
       Selector (field query) to filter on, supports '=', '==', and
       '!='.(e.g. --field-selector key1=value1,key2=value2). The server only
       supports a limited number of field queries per type.

    -f, --filename=[ ]:
       Filename, directory, or URL to files identifying the resource to
       update the annotation

    -k, --kustomize='':
       Process the kustomization directory. This flag can't be used together
```

```
       with -f or -R.

    --list=false:
       If true, display the annotations for a given resource.

    --local=false:
       If true, annotation will NOT contact api-server but run locally.

    -o, --output='':
       Output format. One of: (json, yaml, name, go-template,
       go-template-file, template, templatefile, jsonpath, jsonpath-as-json,
       jsonpath-file).

    --overwrite=false:
       If true, allow annotations to be overwritten, otherwise reject
       annotation updates that overwrite existing annotations.

    -R, --recursive=false:
       Process the directory used in -f, --filename recursively. Useful when
       you want to manage related manifests organized within the same
       directory.

    --resource-version='':
       If non-empty, the annotation update will only succeed if this is the
       current resource-version for the object. Only valid when specifying a
       single resource.

    -l, --selector='':
       Selector (label query) to filter on, supports '=', '==', and
       '!='.(e.g. -l key1=value1,key2=value2). Matching objects must satisfy
       all of the specified label constraints.

    --show-managed-fields=false:
       If true, keep the managedFields when printing objects in JSON or YAML
       format.

    --template='':
       Template string or path to template file to use when -o=go-template,
       -o=go-template-file. The template format is golang templates
       [http://golang.org/pkg/text/template/#pkg-overview].

Usage:
  kubectl annotate [--overwrite] (-f FILENAME | TYPE NAME) KEY_1=VAL_1 ... KEY_N=VAL_N [--
resource-version=version] [options]


================================================================================

kubectl completion --help
Output shell completion code for the specified shell (bash, zsh, fish, or
powershell). The shell code must be evaluated to provide interactive completion
of kubectl commands.  This can be done by sourcing it from the .bash_profile.

 Detailed instructions on how to do this are available here:

  for macOS:

https://kubernetes.io/docs/tasks/tools/install-kubectl-macos/#enable-shell-autocompletion

  for linux:

https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/#enable-shell-autocompletion

  for windows:

https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/#enable-shell-autocompletion
```

```
 Note for zsh users: [1] zsh completions are only supported in versions of zsh
>= 5.2.

Examples:
  # Installing bash completion on macOS using homebrew
  ## If running Bash 3.2 included with macOS
  brew install bash-completion
  ## or, if running Bash 4.1+
  brew install bash-completion@2
  ## If kubectl is installed via homebrew, this should start working immediately
  ## If you've installed via other means, you may need add the completion to your completion
directory
  kubectl completion bash > $(brew --prefix)/etc/bash_completion.d/kubectl


  # Installing bash completion on Linux
  ## If bash-completion is not installed on Linux, install the 'bash-completion' package
  ## via your distribution's package manager.
  ## Load the kubectl completion code for bash into the current shell
  source <(kubectl completion bash)
  ## Write bash completion code to a file and source it from .bash_profile
  kubectl completion bash > ~/.kube/completion.bash.inc printf "
  # Kubectl shell completion
  source '$HOME/.kube/completion.bash.inc'
  " >> $HOME/.bash_profile
  source $HOME/.bash_profile

  # Load the kubectl completion code for zsh[1] into the current shell
  source <(kubectl completion zsh)
  # Set the kubectl completion code for zsh[1] to autoload on startup
  kubectl completion zsh > "${fpath[1]}/_kubectl"


  # Load the kubectl completion code for fish[2] into the current shell
  kubectl completion fish | source
  # To load completions for each session, execute once:
  kubectl completion fish > ~/.config/fish/completions/kubectl.fish

  # Load the kubectl completion code for powershell into the current shell
  kubectl completion powershell | Out-String | Invoke-Expression
  # Set kubectl completion code for powershell to run on startup
  ## Save completion code to a script and execute in the profile
  kubectl completion powershell > $HOME\.kube\completion.ps1
  Add-Content $PROFILE "$HOME\.kube\completion.ps1"
  ## Execute completion code in the profile
  Add-Content $PROFILE "if (Get-Command kubectl -ErrorAction SilentlyContinue) {
  kubectl completion powershell | Out-String | Invoke-Expression
  }"
  ## Add completion code directly to the $PROFILE script
  kubectl completion powershell >> $PROFILE

Usage:
  kubectl completion SHELL [options]

================================================================================

kubectl alpha --help
These commands correspond to alpha features that are not enabled in Kubernetes
clusters by default.

Available Commands:
  auth          Inspect authorization


================================================================================
```

```
kubectl api-resources --help
Print the supported API resources on the server.

Examples:
  # Print the supported API resources
  kubectl api-resources

  # Print the supported API resources with more information
  kubectl api-resources -o wide

  # Print the supported API resources sorted by a column
  kubectl api-resources --sort-by=name

  # Print the supported namespaced resources
  kubectl api-resources --namespaced=true

  # Print the supported non-namespaced resources
  kubectl api-resources --namespaced=false

  # Print the supported API resources with a specific APIGroup
  kubectl api-resources --api-group=rbac.authorization.k8s.io

Options:
    --api-group='':
       Limit to resources in the specified API group.

    --cached=false:
       Use the cached list of resources if available.

    --categories=[ ]:
       Limit to resources that belong the the specified categories.

    --namespaced=true:
       If false, non-namespaced resources will be returned, otherwise
       returning namespaced resources by default.

    --no-headers=false:
       When using the default or custom-column output format, don't print
       headers (default print headers).

    -o, --output='':
       Output format. One of: (wide, name).

    --sort-by='':
       If non-empty, sort list of resources using specified field. The field
       can be either 'name' or 'kind'.

    --verbs=[ ]:
       Limit to resources that support the specified verbs.

Usage:
  kubectl api-resources [flags] [options]


================================================================================

kubectl api-versions --help
Print the supported API versions on the server, in the form of "group/version".

Examples:
  # Print the supported API versions
  kubectl api-versions

Usage:
  kubectl api-versions [options]
```

```
================================================================================

kubectl config --help
Modify kubeconfig files using subcommands like "kubectl config set
current-context my-context"

 The loading order follows these rules:

  1.  If the --kubeconfig flag is set, then only that file is loaded. The flag
may only be set once and no merging takes place.
  2.  If $KUBECONFIG environment variable is set, then it is used as a list of
paths (normal path delimiting rules for your system). These paths are merged.
When a value is modified, it is modified in the file that defines the stanza.
When a value is created, it is created in the first file that exists. If no
files in the chain exist, then it creates the last file in the list.
  3.  Otherwise, ${HOME}/.kube/config is used and no merging takes place.

Available Commands:
  current-context   Display the current-context
  delete-cluster    Delete the specified cluster from the kubeconfig
  delete-context    Delete the specified context from the kubeconfig
  delete-user       Delete the specified user from the kubeconfig
  get-clusters      Display clusters defined in the kubeconfig
  get-contexts      Describe one or many contexts
  get-users         Display users defined in the kubeconfig
  rename-context    Rename a context from the kubeconfig file
  set               Set an individual value in a kubeconfig file
  set-cluster       Set a cluster entry in kubeconfig
  set-context       Set a context entry in kubeconfig
  set-credentials   Set a user entry in kubeconfig
  unset             Unset an individual value in a kubeconfig file
  use-context       Set the current-context in a kubeconfig file
  view              Display merged kubeconfig settings or a specified kubeconfig file

Usage:
  kubectl config SUBCOMMAND [options]



================================================================================

kubectl plugin --help
Provides utilities for interacting with plugins.

 Plugins provide extended functionality that is not part of the major
command-line distribution. Please refer to the documentation and examples for
more information about how write your own plugins.

 The easiest way to discover and install plugins is via the kubernetes
sub-project krew. To install krew, visit
https://krew.sigs.k8s.io/docs/user-guide/setup/install/ krew.sigs.k8s.io
https://krew.sigs.k8s.io/docs/user-guide/setup/install/

Available Commands:
  list          List all visible plugin executables on a user's PATH

Usage:
  kubectl plugin [flags] [options]


================================================================================

kubectl version --help
Print the client and server version information for the current context.
```

```
Examples:
  # Print the client and server versions for the current context
  kubectl version

Options:
    --client=false:
       If true, shows client version only (no server required).

    -o, --output='':
       One of 'yaml' or 'json'.

Usage:
  kubectl version [flags] [options]


================================================================================




Password:
[root@freeipa tm]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
[root@freeipa tm]# dnf install -y kubelet kubeadm kubectl
[sudo] password for tm:
Extra Packages for Enterprise Linux 9 - x86_64   24 kB/s |  29 kB     00:01
Extra Packages for Enterprise Linux 9 - x86_64  408 kB/s |  15 MB     00:36
Kubernetes                                      177  B/s | 454  B     00:02
Kubernetes                                       11 kB/s | 2.6 kB     00:00
Importing GPG key 0x13EDEF05:
 Userid     : "Rapture Automatic Signing Key (cloud-rapture-signing-key-2022-03-07-
08_01_01.pub)"
 Fingerprint: A362 B822 F6DE DC65 2817 EA46 B53D C80D 13ED EF05
 From       : https://packages.cloud.google.com/yum/doc/yum-key.gpg
Importing GPG key 0xDC6315A3:
 Userid     : "Artifact Registry Repository Signer <artifact-registry-repository-
signer@google.com>"
 Fingerprint: 35BA A0B3 3E9E B396 F59C A838 C0BA 5CE6 DC63 15A3
 From       : https://packages.cloud.google.com/yum/doc/yum-key.gpg
Kubernetes                                      3.2 kB/s | 975  B     00:00
Importing GPG key 0x3E1BA8D5:
 Userid     : "Google Cloud Packages RPM Signing Key <gc-team@google.com>"
 Fingerprint: 3749 E1BA 95A8 6CE0 5454 6ED2 F09C 394C 3E1B A8D5
 From       : https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
Kubernetes                                       98 kB/s | 168 kB     00:01
Rocky Linux 9 - BaseOS                          5.7 kB/s | 4.1 kB     00:00
Rocky Linux 9 - BaseOS                          279 kB/s | 1.8 MB     00:06
Rocky Linux 9 - AppStream                       5.8 kB/s | 4.5 kB     00:00
Rocky Linux 9 - AppStream                       275 kB/s | 6.6 MB     00:24
Rocky Linux 9 - Extras                          505  B/s | 2.9 kB     00:05
Rocky Linux 9 - Extras                           12 kB/s | 8.5 kB     00:00
RPM Fusion for EL 9 - Free - Updates            8.7 kB/s | 2.8 kB     00:00
RPM Fusion for EL 9 - Free - Updates            160 kB/s | 245 kB     00:01
RPM Fusion for EL 9 - Nonfree - Updates          34 kB/s | 5.6 kB     00:00
RPM Fusion for EL 9 - Nonfree - Updates          66 kB/s |  63 kB     00:00
```

```
Dependencies resolved.
================================================================================
 Package                    Arch       Version              Repository      Size
================================================================================
Installing:
 kubeadm                    x86_64     1.26.3-0             kubernetes      10 M
 kubectl                    x86_64     1.26.3-0             kubernetes      11 M
 kubelet                    x86_64     1.26.3-0             kubernetes      22 M
Installing dependencies:
 conntrack-tools            x86_64     1.4.5-17.el9_1       appstream      210 k
 cri-tools                  x86_64     1.26.0-0             kubernetes     8.6 M
 kubernetes-cni             x86_64     1.2.0-0              kubernetes      17 M
 libnetfilter_cthelper      x86_64     1.0.0-22.el9         appstream       23 k
 libnetfilter_cttimeout     x86_64     1.0.0-19.el9         appstream       23 k
 libnetfilter_queue         x86_64     1.0.5-1.el9          appstream       28 k
 socat                      x86_64     1.7.4.1-5.el9        appstream      300 k

Transaction Summary
================================================================================
Install  10 Packages

Total download size: 69 M
Installed size: 296 M
Downloading Packages:
(1/10): 3f5ba2b53701ac9102ea7c7ab2ca6616a8cd596 117 kB/s | 8.6 MB     01:15
(2/10): 6dee5ef05a942f42f7e49d2aac919d1177d8006 129 kB/s |  10 MB     01:22
(3/10): 8c423fd76dc6d7a5dfc2aa30630cb46b6bd8128 114 kB/s |  11 MB     01:35
(4/10): conntrack-tools-1.4.5-17.el9_1.x86_64.r 130 kB/s | 210 kB     00:01
(5/10): libnetfilter_cttimeout-1.0.0-19.el9.x86 103 kB/s |  23 kB     00:00
(6/10): socat-1.7.4.1-5.el9.x86_64.rpm          178 kB/s | 300 kB     00:01
(7/10): libnetfilter_cthelper-1.0.0-22.el9.x86_  77 kB/s |  23 kB     00:00
(8/10): libnetfilter_queue-1.0.5-1.el9.x86_64.r 108 kB/s |  28 kB     00:00
(9/10): 0f2a2afd740d476ad77c508847bad1f559afc24 214 kB/s |  17 MB     01:20
(10/10): 4bd2c321343cbe55e4e50e6095273a2bca4ecd 233 kB/s |  22 MB     01:35
--------------------------------------------------------------------------------
Total                                           411 kB/s |  69 MB     02:51
Kubernetes                                      8.9 kB/s | 2.6 kB     00:00
Importing GPG key 0x13EDEF05:
 Userid     : "Rapture Automatic Signing Key (cloud-rapture-signing-key-2022-03-07-
08_01_01.pub)"
 Fingerprint: A362 B822 F6DE DC65 2817 EA46 B53D C80D 13ED EF05
 From       : https://packages.cloud.google.com/yum/doc/yum-key.gpg
Key imported successfully
Importing GPG key 0xDC6315A3:
 Userid     : "Artifact Registry Repository Signer <artifact-registry-repository-
signer@google.com>"
 Fingerprint: 35BA A0B3 3E9E B396 F59C A838 C0BA 5CE6 DC63 15A3
 From       : https://packages.cloud.google.com/yum/doc/yum-key.gpg
Key imported successfully
Kubernetes                                      3.9 kB/s | 975  B     00:00
Importing GPG key 0x3E1BA8D5:
 Userid     : "Google Cloud Packages RPM Signing Key <gc-team@google.com>"
 Fingerprint: 3749 E1BA 95A8 6CE0 5454 6ED2 F09C 394C 3E1B A8D5
 From       : https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : libnetfilter_queue-1.0.5-1.el9.x86_64                  1/10
  Installing       : libnetfilter_cthelper-1.0.0-22.el9.x86_64              2/10
  Installing       : socat-1.7.4.1-5.el9.x86_64                             3/10
  Installing       : libnetfilter_cttimeout-1.0.0-19.el9.x86_64             4/10
  Installing       : conntrack-tools-1.4.5-17.el9_1.x86_64                  5/10
```

```
  Running scriptlet: conntrack-tools-1.4.5-17.el9_1.x86_64                  5/10
  Installing       : kubernetes-cni-1.2.0-0.x86_64                          6/10
  Installing       : kubelet-1.26.3-0.x86_64                                7/10
  Installing       : kubectl-1.26.3-0.x86_64                                8/10
  Installing       : cri-tools-1.26.0-0.x86_64                              9/10
  Installing       : kubeadm-1.26.3-0.x86_64                               10/10
  Running scriptlet: kubeadm-1.26.3-0.x86_64                               10/10
  Verifying        : cri-tools-1.26.0-0.x86_64                              1/10
  Verifying        : kubeadm-1.26.3-0.x86_64                                2/10
  Verifying        : kubectl-1.26.3-0.x86_64                                3/10
  Verifying        : kubelet-1.26.3-0.x86_64                                4/10
  Verifying        : kubernetes-cni-1.2.0-0.x86_64                          5/10
  Verifying        : conntrack-tools-1.4.5-17.el9_1.x86_64                  6/10
  Verifying        : libnetfilter_cttimeout-1.0.0-19.el9.x86_64             7/10
  Verifying        : socat-1.7.4.1-5.el9.x86_64                             8/10
  Verifying        : libnetfilter_cthelper-1.0.0-22.el9.x86_64              9/10
  Verifying        : libnetfilter_queue-1.0.5-1.el9.x86_64                 10/10

Installed:
  conntrack-tools-1.4.5-17.el9_1.x86_64
  cri-tools-1.26.0-0.x86_64
  kubeadm-1.26.3-0.x86_64
  kubectl-1.26.3-0.x86_64
  kubelet-1.26.3-0.x86_64
  kubernetes-cni-1.2.0-0.x86_64
  libnetfilter_cthelper-1.0.0-22.el9.x86_64
  libnetfilter_cttimeout-1.0.0-19.el9.x86_64
  libnetfilter_queue-1.0.5-1.el9.x86_64
  socat-1.7.4.1-5.el9.x86_64
```