

- **GETTING STARTED**

- [create](#)
- [get](#)
- [run](#)
- [expose](#)
- [delete](#)

- **APP MANAGEMENT**

- [apply](#)
- [annotate](#)
- [autoscale](#)
- [debug](#)
- [diff](#)
- [edit](#)
- [kustomize](#)
- [label](#)
- [patch](#)
- [replace](#)
- [rollout](#)
- [scale](#)
- [set](#)
- [wait](#)

- **WORKING WITH APPS**

- [attach](#)
- [auth](#)
- [cp](#)
- [describe](#)
- [exec](#)
- [logs](#)

- [port-forward](#)
- [proxy](#)
- [top](#)
- [CLUSTER MANAGEMENT](#)
- [api-versions](#)
- [certificate](#)
- [cluster-info](#)
- [cordon](#)
- [drain](#)
- [taint](#)
- [uncordon](#)
- [KUBECTL SETTINGS AND USAGE](#)
- [alpha](#)
- [api-resources](#)
- [completion](#)
- [config](#)
- [explain](#)
- [options](#)
- [plugin](#)
- [version](#)

[Copyright 2020 The Kubernetes Authors.](#)

- [example](#)

GETTING STARTED

This section contains the most basic commands for getting a workload running on your cluster.

- [run](#) will start running 1 or more instances of a container image on your cluster.
- [expose](#) will load balance traffic across the running instances, and can create a HA proxy for accessing the containers from outside the cluster.

Once your workloads are running, you can use the commands in the [WORKING WITH APPS](#) section to inspect them.

create

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 docker-registry.yaml in JSON then create the resource using the edited data

```
kubectl create -f docker-registry.yaml --edit -o json
```

Create a resource from a file or from stdin.

JSON and YAML formats are accepted.

Usage

```
$ kubectl create -f FILENAME
```

Flags

Name	Shorthand	Default	Usage
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 goyaml 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.
filename	f	[]	Filename, directory, or URL to files to use to create the resource
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		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.

Name	Shorthand	Default	Usage
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	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	l		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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it
windows-line-endings		false	Only relevant if --edit=true. Defaults to the line ending native to your platform.

clusterrole

Create a cluster role named "pod-reader" that allows user to perform "get", "watch" and "list" on pods

```
kubectl create clusterrole pod-reader --verb=get,list,watch --resource=pods
```

Create a cluster role named "pod-reader" with ResourceName specified

```
kubectl create clusterrole pod-reader --verb=get --resource=pods --resource-name=readablepod --resource-name=anotherpod
```

Create a cluster role named "foo" with API Group specified

```
kubectl create clusterrole foo --verb=get,list,watch --resource=rs.extensions
```

Create a cluster role named "foo" with SubResource specified

```
kubectl create clusterrole foo --verb=get,list,watch --resource=pods,pods/status
```

Create a cluster role name "foo" with NonResourceURL specified

```
kubectl create clusterrole "foo" --verb=get --non-resource-url=/logs/*
```

Create a cluster role name "monitoring" with AggregationRule specified

```
kubectl create clusterrole monitoring --aggregation-rule="rbac.example.com/aggregate-to-monitoring=true"
```

Create a cluster role.

Usage

```
$ kubectl create clusterrole NAME --verb=verb --resource=resource.group [--resource-name=resourcename] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
aggregation-rule			An aggregation label selector for combining ClusterRoles.
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-create	Name of the manager used to track field ownership.
non-resource-url		[]	A partial url that user should have access to.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
resource		[]	Resource that the rule applies to
resource-name		[]	Resource in the white list that the rule applies to, repeat this flag for multiple items
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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview] .
validate		true	If true, use a schema to validate the input before sending it
verb		[]	Verb that applies to the resources contained in the rule

clusterrolebinding

Create a cluster role binding for user1, user2, and group1 using the cluster-admin cluster role

```
kubectl create clusterrolebinding cluster-admin --clusterrole=cluster-admin --user=user1 --user=user2 --group=group1
```

Create a cluster role binding for a particular cluster role.

Usage

```
$ kubectl create clusterrolebinding NAME --clusterrole=NAME [--user=username] [--group=groupname] [--serviceaccount=namespace:serviceaccountname] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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.
clusterrole			ClusterRole this ClusterRoleBinding should reference
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-create	Name of the manager used to track field ownership.
group		[]	Groups to bind to the clusterrole
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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.
serviceaccount		[]	Service accounts to bind to the clusterrole, in the format <namespace>:<name>
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		true	If true, use a schema to validate the input before sending it

configmap

Create a new config map named my-config based on folder bar

```
kubectl create configmap my-config --from-file=path/to/bar
```

Create a new config map named my-config with specified keys instead of file basenames on disk

```
kubectl create configmap my-config --from-file=key1=/path/to/bar/file1.txt --from-file=key2=/path/to/bar/file2.txt
```

Create a new config map named my-config with key1=config1 and key2=config2

```
kubectl create configmap my-config --from-literal=key1=config1 --from-literal=key2=config2
```

Create a new config map named my-config from the key=value pairs in the file

```
kubectl create configmap my-config --from-file=path/to/bar
```

Create a new config map named my-config from an env file

```
kubectl create configmap my-config --from-env-file=path/to/bar.env
```

Create a config map based on a file, directory, or specified literal value.

A single config map may package one or more key/value pairs.

When creating a config map based on a file, the key will default to the basename of the file, and the value will default to the file content. If the basename is an invalid key, you may specify an alternate key.

When creating a config map based on a directory, each file whose basename is a valid key in the directory will be packaged into the config map. Any directory entries except regular files are ignored (e.g. subdirectories, symlinks, devices, pipes, etc).

Usage

```
$ kubectl create configmap NAME [--from-file=[key=]source] [--from-literal=key1=value1] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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 goyaml and jsonpath output formats.
append-hash		false	Append a hash of the configmap to its name.
dry-run		none	Must be "none", "server", or "client". If client strategy, only print the object that would be sent, without sending it. If

Name	Shorthand	Default	Usage
			server strategy, submit server-side request without persisting the resource.
field-manager		kubectl-create	Name of the manager used to track field ownership.
from-env-file			Specify the path to a file to read lines of key=val pairs to create a configmap (i.e. a Docker .env file).
from-file		[]	Key file can be specified using its file path, in which case file basename will be used as configmap key, or optionally with a key and file path, in which case the given key will be used. Specifying a directory will iterate each named file in the directory whose basename is a valid configmap key.
from-literal		[]	Specify a key and literal value to insert in configmap (i.e. mykey=somevalue)
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

cronjob

Create a cron job

```
kubectl create cronjob my-job --image=busybox --schedule="*/1 * * * *"
```

Create a cron job with a command

```
kubectl create cronjob my-job --image=busybox --schedule="*/1 * * * *" -- date
```

Create a cron job with the specified name.

Usage

```
$ kubectl create cronjob NAME --image=image --schedule='0/5 * * * ?' -- [COMMAND] [args...]
```


Flags

Name	Shorthand	Default	Usage
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 goolang 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-create	Name of the manager used to track field ownership.
image			Image name to run.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
restart			job's restart policy. supported values: OnFailure, Never
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.
schedule			A schedule in the Cron format the job should be run with.
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 goolang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

deployment

Create a deployment named my-dep that runs the busybox image

```
kubectl create deployment my-dep --image=busybox
```

Create a deployment with a command

```
kubectl create deployment my-dep --image=busybox -- date
```

Create a deployment named my-dep that runs the nginx image with 3 replicas

```
kubectl create deployment my-dep --image=nginx --replicas=3
```

Create a deployment named my-dep that runs the busybox image and expose port 5701

```
kubectl create deployment my-dep --image=busybox --port=5701
```

Create a deployment with the specified name.

Usage

```
$ kubectl create deployment NAME --image=image -- [COMMAND] [args...]
```

Flags

Name	Shorthand	Default	Usage
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 goolang 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-create	Name of the manager used to track field ownership.
image		[]	Image names to run.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
port		-1	The port that this container exposes.
replicas	r	1	Number of replicas to create. Default is 1.
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 goolang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

ingress

Create a single ingress called 'simple' that directs requests to foo.com/bar to svc # svc1:8080 with a tls secret "my-cert"

```
kubectl create ingress simple --rule="foo.com/bar=svc1:8080,tls=my-cert"
```

Create a catch all ingress of "/path" pointing to service svc:port and Ingress Class as "otheringress"

```
kubectl create ingress catch-all --class=otheringress --rule="/path=svc:port"
```

Create an ingress with two annotations: ingress.annotation1 and ingress.annotations2

```
kubectl create ingress annotated --class=default --rule="foo.com/bar=svc:port" \
--annotation ingress.annotation1=foo \
--annotation ingress.annotation2=bla
```

Create an ingress with the same host and multiple paths

```
kubectl create ingress multipath --class=default \
--rule="foo.com/=svc:port" \
--rule="foo.com/admin/=svcadmin:portadmin"
```

Create an ingress with multiple hosts and the pathType as Prefix

```
kubectl create ingress ingress1 --class=default \
--rule="foo.com/path*=svc:8080" \
--rule="bar.com/admin*=svc2:http"
```

Create an ingress with TLS enabled using the default ingress certificate and different path types

```
kubectl create ingress ingtls --class=default \
--rule="foo.com/=svc:https,tls" \
--rule="foo.com/path/subpath*=othersvc:8080"
```

Create an ingress with TLS enabled using a specific secret and pathType as Prefix

```
kubectl create ingress ingsecret --class=default \
--rule="foo.com/*=svc:8080,tls=secret1"
```

Create an ingress with a default backend

```
kubectl create ingress ingdefault --class=default \
--default-backend=defaultsvc:http \
--rule="foo.com/*=svc:8080,tls=secret1"
```

Create an ingress with the specified name.

Usage

```
$ kubectl create ingress NAME --rule=host/path=service:port[,tls[=secret]]
```

Flags

Name	Shorthand	Default	Usage
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.
annotation		[]	Annotation to insert in the ingress object, in the format annotation=value
class			Ingress Class to be used

Name	Shorthand	Default	Usage
default-backend			Default service for backend, in format of svcname:port
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-create	Name of the manager used to track field ownership.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
rule		[]	Rule in format host/path=service:port[,tls=secretname]. Paths containing the leading character '*' are considered pathType=Prefix. tls argument is optional.
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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

job

Create a job

```
kubectl create job my-job --image=busybox
```

Create a job with a command

```
kubectl create job my-job --image=busybox -- date
```

Create a job from a cron job named "a-cronjob"

```
kubectl create job test-job --from=cronjob/a-cronjob
```

Create a job with the specified name.

Usage

```
$ kubectl create job NAME --image=image [--from=cronjob/name] -- [COMMAND] [args...]
```

Flags

Name	Shorthand	Default	Usage
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-create	Name of the manager used to track field ownership.
from			The name of the resource to create a Job from (only cronjob is supported).
image			Image name to run.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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].
validate		true	If true, use a schema to validate the input before sending it

namespace

Create a new namespace named my-namespace

```
kubectl create namespace my-namespace
```

Create a namespace with the specified name.

Usage

```
$ kubectl create namespace NAME [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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.

Name	Shorthand	Default	Usage
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-create	Name of the manager used to track field ownership.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

poddisruptionbudget

Create a pod disruption budget named my-pdb that will select all pods with the app=rails label # and require at least one of them being available at any point in time

```
kubectl create poddisruptionbudget my-pdb --selector=app=rails --min-available=1
```

Create a pod disruption budget named my-pdb that will select all pods with the app=nginx label # and require at least half of the pods selected to be available at any point in time

```
kubectl create pdb my-pdb --selector=app=nginx --min-available=50%
```

Create a pod disruption budget with the specified name, selector, and desired minimum available pods.

Usage

```
$ kubectl create poddisruptionbudget NAME --selector=SELECTOR --min-available=N [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
allow-missing-template-keys		true	

Name	Shorthand	Default	Usage
			If true, ignore any errors in templates when a field or map key is missing in the template. Only applies to go-lang 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-create	Name of the manager used to track field ownership.
max-unavailable			The maximum number or percentage of unavailable pods this budget requires.
min-available			The minimum number or percentage of available pods this budget requires.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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 budget. Only equality-based selector requirements are supported.
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 go-lang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

priorityclass

Create a priority class named high-priority

```
kubectl create priorityclass high-priority --value=1000 --description="high priority"
```

Create a priority class named default-priority that is considered as the global default priority

```
kubectl create priorityclass default-priority --value=1000 --global-default=true --description="default priority"
```

Create a priority class named high-priority that cannot preempt pods with lower priority

```
kubectl create priorityclass high-priority --value=1000 --description="high priority" --preemption-policy="Never"
```

Create a priority class with the specified name, value, globalDefault and description.

Usage

```
$ kubectl create priorityclass NAME --value=VALUE --global-default=BOOL [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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 go lang and jsonpath output formats.
description			description is an arbitrary string that usually provides guidelines on when this priority class should 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-create	Name of the manager used to track field ownership.
global-default		false	global-default specifies whether this PriorityClass should be considered as the default priority.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
preemption-policy		PreemptLowerPriority	preemption-policy is the policy for preempting pods with lower priority.
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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it
value		0	the value of this priority class.

quota

Create a new resource quota named my-quota

```
kubectl create quota my-quota --hard=cpu=1,memory=1G,pods=2,services=3,replicationcontrollers=2,resourcequotas=1,secrets=5,persistentvolumeclaims=10
```

Create a new resource quota named best-effort

```
kubectl create quota best-effort --hard=pods=100 --scopes=BestEffort
```

Create a resource quota with the specified name, hard limits, and optional scopes.

Usage

```
$ kubectl create quota NAME [--hard=key1=value1,key2=value2] [--scopes=Scope1,Scope2] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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-create	Name of the manager used to track field ownership.
hard			A comma-delimited set of resource=quantity pairs that define a hard limit.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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.
scopes			A comma-delimited set of quota scopes that must all match each object tracked by the quota.
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		true	If true, use a schema to validate the input before sending it

role

Create a role named "pod-reader" that allows user to perform "get", "watch" and "list" on pods

```
kubectl create role pod-reader --verb=get --verb=list --verb=watch --resource=pods
```

Create a role named "pod-reader" with ResourceName specified

```
kubectl create role pod-reader --verb=get --resource=pods --resource-name=readablepod --resource-name=anotherpod
```

Create a role named "foo" with API Group specified

```
kubectl create role foo --verb=get,list,watch --resource=rs.extensions
```

Create a role named "foo" with SubResource specified

```
kubectl create role foo --verb=get,list,watch --resource=pods,pods/status
```

Create a role with single rule.

Usage

```
$ kubectl create role NAME --verb=verb --resource=resource.group/subresource [--resource-name=resourcename] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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-create	Name of the manager used to track field ownership.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
resource		[]	Resource that the rule applies to
resource-name		[]	Resource in the white list that the rule applies to, repeat this flag for multiple items
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.
		false	If true, keep the managedFields when printing objects in JSON or YAML format.

Name	Shorthand	Default	Usage
show-managed-fields			
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is goolang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it
verb		[]	Verb that applies to the resources contained in the rule

rolebinding

Create a role binding for user1, user2, and group1 using the admin cluster role

```
kubectl create rolebinding admin --clusterrole=admin --user=user1 --user=user2 --group=group1
```

Create a role binding for a particular role or cluster role.

Usage

```
$ kubectl create rolebinding NAME --clusterrole=NAME|--role=NAME [--user=username] [--group=groupname] [--serviceaccount=namespace:serviceaccountname] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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 goolang and jsonpath output formats.
clusterrole			ClusterRole this RoleBinding should reference
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-create	Name of the manager used to track field ownership.
group		[]	Groups to bind to the role
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
role			Role this RoleBinding should reference
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.
serviceaccount		[]	

Name	Shorthand	Default	Usage
			Service accounts to bind to the role, in the format <namespace>:<name>
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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview] .
validate		true	If true, use a schema to validate the input before sending it

secret

Create a secret using specified subcommand.

Usage

```
$ kubectl create secret
```

secret docker-registry

If you don't already have a .dockercfg file, you can create a dockercfg secret directly by using:

```
kubectl create secret docker-registry my-secret --docker-server=DOCKER_REGISTRY_SERVER
--docker-username=DOCKER_USER --docker-password=DOCKER_PASSWORD --docker-
email=DOCKER_EMAIL
```

Create a new secret named my-secret from ~/.docker/config.json

```
kubectl create secret docker-registry my-secret --from-file=.dockerconfigjson=path/to/.docker/
config.json
```

Create a new secret for use with Docker registries.

Dockercfg secrets are used to authenticate against Docker registries.

When using the Docker command line to push images, you can authenticate to a given registry by running: '\$ docker login DOCKER_REGISTRY_SERVER --username=DOCKER_USER --password=DOCKER_PASSWORD --email=DOCKER_EMAIL'.

That produces a ~/.dockercfg file that is used by subsequent 'docker push' and 'docker pull' commands to authenticate to the registry. The email address is optional.

When creating applications, you may have a Docker registry that requires authentication. In order for the nodes to pull images on your behalf, they must have the credentials. You can provide this information by creating a dockercfg secret and attaching it to your service account.

Usage

```
$ kubectl create secret docker-registry NAME --docker-username=user --docker-  
password=password --docker-email=email [--docker-server=string] [--from-file=[key=]source]  
[--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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 goyaml and jsonpath output formats.
append-hash		false	Append a hash of the secret to its name.
docker-email			Email for Docker registry
docker-password			Password for Docker registry authentication
docker-server		https:// index.docker.io/ v1/	Server location for Docker registry
docker-username			Username for Docker registry authentication
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-create	Name of the manager used to track field ownership.
from-file		[]	Key files can be specified using their file path, in which case a default name will be given to them, or optionally with a name and file path, in which case the given name will be used. Specifying a directory will iterate each named file in the directory that is a valid secret key.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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

Name	Shorthand	Default	Usage
			format is go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

secret generic

Create a new secret named my-secret with keys for each file in folder bar

```
kubectl create secret generic my-secret --from-file=path/to/bar
```

Create a new secret named my-secret with specified keys instead of names on disk

```
kubectl create secret generic my-secret --from-file=ssh-privatekey=path/to/id_rsa --from-file=ssh-publickey=path/to/id_rsa.pub
```

Create a new secret named my-secret with key1=supersecret and key2=topsecret

```
kubectl create secret generic my-secret --from-literal=key1=supersecret --from-literal=key2=topsecret
```

Create a new secret named my-secret using a combination of a file and a literal

```
kubectl create secret generic my-secret --from-file=ssh-privatekey=path/to/id_rsa --from-literal=passphrase=topsecret
```

Create a new secret named my-secret from an env file

```
kubectl create secret generic my-secret --from-env-file=path/to/bar.env
```

Create a secret based on a file, directory, or specified literal value.

A single secret may package one or more key/value pairs.

When creating a secret based on a file, the key will default to the basename of the file, and the value will default to the file content. If the basename is an invalid key or you wish to choose your own, you may specify an alternate key.

When creating a secret based on a directory, each file whose basename is a valid key in the directory will be packaged into the secret. Any directory entries except regular files are ignored (e.g. subdirectories, symlinks, devices, pipes, etc).

Usage

```
$ kubectl create generic NAME [--type=string] [--from-file=[key=]source] [--from-literal=key1=value1] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
		true	

Name	Shorthand	Default	Usage
allow-missing-template-keys			If true, ignore any errors in templates when a field or map key is missing in the template. Only applies to go lang and jsonpath output formats.
append-hash		false	Append a hash of the secret to its name.
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-create	Name of the manager used to track field ownership.
from-env-file			Specify the path to a file to read lines of key=val pairs to create a secret (i.e. a Docker .env file).
from-file		[]	Key files can be specified using their file path, in which case a default name will be given to them, or optionally with a name and file path, in which case the given name will be used. Specifying a directory will iterate each named file in the directory that is a valid secret key.
from-literal		[]	Specify a key and literal value to insert in secret (i.e. mykey=somevalue)
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
type			The type of secret to create
validate		true	If true, use a schema to validate the input before sending it

secret tls

Create a new TLS secret named tls-secret with the given key pair

```
kubectl create secret tls tls-secret --cert=path/to/tls.cert --key=path/to/tls.key
```

Create a TLS secret from the given public/private key pair.

The public/private key pair must exist beforehand. The public key certificate must be .PEM encoded and match the given private key.

Usage

```
$ kubectl create secret tls NAME --cert=path/to/cert/file --key=path/to/key/file [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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.
append-hash		false	Append a hash of the secret to its name.
cert			Path to PEM encoded public key certificate.
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-create	Name of the manager used to track field ownership.
key			Path to private key associated with given certificate.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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].
validate		true	If true, use a schema to validate the input before sending it

service

Create a service using a specified subcommand.

Usage

```
$ kubectl create service
```

service clusterip

Create a new ClusterIP service named my-cs


```
kubectl create service clusterip my-cs --tcp=5678:8080
```

Create a new ClusterIP service named my-cs (in headless mode)

```
kubectl create service clusterip my-cs --clusterip="None"
```

Create a ClusterIP service with the specified name.

Usage

```
$ kubectl create service clusterip NAME [--tcp=<port>:<targetPort>] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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.
clusterip			Assign your own ClusterIP or set to 'None' for a 'headless' service (no loadbalancing).
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-create	Name of the manager used to track field ownership.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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.
tcp		[]	Port pairs can be specified as '<port>:<targetPort>'.
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		true	If true, use a schema to validate the input before sending it

service externalname

Create a new ExternalName service named my-ns

```
kubectl create service externalname my-ns --external-name bar.com
```

Create an ExternalName service with the specified name.

ExternalName service references to an external DNS address instead of only pods, which will allow application authors to reference services that exist off platform, on other clusters, or locally.

Usage

```
$ kubectl create service externalname NAME --external-name external.name [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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 goolang 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.
external-name			External name of service
field-manager		kubectl-create	Name of the manager used to track field ownership.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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.
tcp		[]	Port pairs can be specified as '<port>:<targetPort>'.
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is goolang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

service loadbalancer

Create a new LoadBalancer service named my-lbs

```
kubectl create service loadbalancer my-lbs --tcp=5678:8080
```

Create a LoadBalancer service with the specified name.

Usage

```
$ kubectl create service loadbalancer NAME [--tcp=port:targetPort] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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 goolang 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-create	Name of the manager used to track field ownership.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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.
tcp		[]	Port pairs can be specified as '<port>:<targetPort>'.
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is goolang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

service nodeport

Create a new NodePort service named my-ns

```
kubectl create service nodeport my-ns --tcp=5678:8080
```

Create a NodePort service with the specified name.

Usage

```
$ kubectl create service nodeport NAME [--tcp=port:targetPort] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
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-create	Name of the manager used to track field ownership.
node-port		0	Port used to expose the service on each node in a cluster.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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.
tcp		[]	Port pairs can be specified as '<port>:<targetPort>'.
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		true	If true, use a schema to validate the input before sending it

serviceaccount

Create a new service account named my-service-account

```
kubectl create serviceaccount my-service-account
```

Create a service account with the specified name.

Usage

\$ kubectl create serviceaccount NAME [--dry-run=server|client|none]

Flags

Name	Shorthand	Default	Usage
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.

Name	Shorthand	Default	Usage
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-create	Name of the manager used to track field ownership.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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].
validate		true	If true, use a schema to validate the input before sending it

get

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
```

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.

Uninitialized objects are not shown unless --include-uninitialized is passed.

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.

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]
```

Flags

Name	Shorthand	Default	Usage
all-namespaces	A	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 go lang 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.

Name	Shorthand	Default	Usage
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.
filename	f	[]	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.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
label-columns	L	[]	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).
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file custom-columns-file custom-columns wide See custom columns [https://kubernetes.io/docs/reference/kubectl/overview/#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.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
selector	l		Selector (label query) to filter on, supports '=', '==', and '!='. (e.g. -l key1=value1,key2=value2)
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.
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].
		false	

Name	Shorthand	Default	Usage
use-openapi-print-columns			If true, use x-kubernetes-print-column metadata (if present) from the OpenAPI schema for displaying a resource.
watch	w	false	After listing/getting the requested object, watch for changes. Uninitialized objects are excluded if no object name is provided.
watch-only		false	Watch for changes to the requested object(s), without listing/getting first.

run

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>
```


Create and run a particular image in a pod.

Usage

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

Flags

Name	Shorthand	Default	Usage
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.
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.
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, service is created for the container(s) which are run
field-manager		kubectl-run	Name of the manager used to track field ownership.
filename	f	[]	to use to replace the resource.
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).
hostport		-1	The host port mapping for the container port. To demonstrate a single-machine container.
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
kustomize	k		

Name	Shorthand	Default	Usage
			Process a kustomization directory. This flag can't be used together with -f or -R.
labels	l		Comma separated labels to apply to the pod(s). 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.
limits			The resource requirement limits for this container. For example, 'cpu=200m,memory=512Mi'. Note that server side components may assign limits depending on the server configuration, such as limit ranges.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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.
quiet	q	false	If true, suppress prompt messages.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
requests			The resource requirement requests for this container. For example, 'cpu=100m,memory=256Mi'. Note that server side components may assign requests depending on the server configuration, such as limit ranges.
restart		Always	The restart policy for this Pod. Legal values [Always, OnFailure, Never].
rm		false	If true, delete resources created in this command for attached containers.
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.
serviceaccount			Service account to set in the pod spec.
show-managed-fields		false	If true, keep the managedFields when printing objects in JSON or YAML format.
stdin	i	false	

Name	Shorthand	Default	Usage
			Keep stdin open on the container(s) 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].
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
tty	t	false	Allocated a TTY for each container in the pod.
wait		false	If true, wait for resources to be gone before returning. This waits for finalizers.

expose

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 to the 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
```

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)

Usage

```
$ kubectrl 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]
```

Flags

Name	Shorthand	Default	Usage
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 goyaml 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.
container-port			Synonym for --target-port
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		kubectrl-expose	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files identifying the resource to expose a service
generator		service/v2	The name of the API generator to use. There are 2 generators: 'service/v1' and 'service/v2'. The only difference between them is that service port in v1 is named 'default', while it is left unnamed in v2. Default is 'service/v2'.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
labels	l		Labels to apply to the service created by this call.

Name	Shorthand	Default	Usage
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.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
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'.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
type			Type for this service: ClusterIP, NodePort, LoadBalancer, or ExternalName. Default is 'ClusterIP'.

delete

Delete a pod using the type and name specified in pod.json

```
kubectrl delete -f ./pod.json
```

Delete resources from a directory containing kustomization.yaml - e.g. dir/
kustomization.yaml

```
kubectrl delete -k dir
```

Delete a pod based on the type and name in the JSON passed into stdin

```
cat pod.json | kubectrl delete -f -
```

Delete pods and services with same names "baz" and "foo"

```
kubectrl delete pod,service baz foo
```

Delete pods and services with label name=myLabel

```
kubectrl delete pods,services -l name=myLabel
```

Delete a pod with minimal delay

```
kubectrl delete pod foo --now
```

Force delete a pod on a dead node

```
kubectrl delete pod foo --force
```

Delete all pods

```
kubectrl delete pods --all
```

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.

Usage

```
$ kubectl delete ([-f FILENAME] | [-k DIRECTORY] | TYPE [(NAME | -l label | --all)])
```

Flags

Name	Shorthand	Default	Usage
all		false	Delete all resources, including uninitialized ones, in the namespace of the specified resource types.
all-namespaces	A	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.
filename	f	[]	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.
kustomize	k		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).
output	o		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.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.

Name	Shorthand	Default	Usage
selector	l		Selector (label query) to filter on, not including uninitialized ones.
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.

APP MANAGEMENT

This section contains commands for creating, updating, deleting, and viewing your workloads in a Kubernetes cluster.

apply

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 -
```

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-whitelist=core/v1/ConfigMap
```

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>.

Usage

\$ kubectl apply (-f FILENAME | -k DIRECTORY)

Flags

Name	Shorthand	Default	Usage
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 goyaml 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.
filename	f	[]	that contains the configuration 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).
kustomize	k		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.
output	o		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, including the uninitialized ones, 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.

Name	Shorthand	Default	Usage
prune-whitelist		[]	Overwrite the default whitelist with <group/version/kind> for --prune
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
selector	l		Selector (label query) to filter on, supports '=', '==', and '!='.(e.g. -l key1=value1,key2=value2)
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 go lang 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		true	If true, use a schema to validate the input before sending it
wait		false	If true, wait for resources to be gone before returning. This waits for finalizers.

edit-last-applied

Edit the last-applied-configuration annotations by type/name in YAML

```
kubectl apply edit-last-applied deployment/nginx
```

Edit the last-applied-configuration annotations by file in JSON

```
kubectl apply edit-last-applied -f deploy.yaml -o json
```

Edit the latest last-applied-configuration annotations of resources from the default editor.

The edit-last-applied 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.

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.

Usage

```
$ kubectl apply edit-last-applied (RESOURCE/NAME | -f FILENAME)
```

Flags

Name	Shorthand	Default	Usage
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 goyang and jsonpath output formats.
field-manager		kubectl-client-side-apply	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files to use to edit the resource
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	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.
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is goyang templates [http://golang.org/pkg/text/template/#pkg-overview].
windows-line-endings		false	Defaults to the line ending native to your platform.

set-last-applied

Set the last-applied-configuration of a resource to match the contents of a file

```
kubectl apply set-last-applied -f deploy.yaml
```

Execute set-last-applied against each configuration file in a directory

```
kubectl apply set-last-applied -f path/
```

Set the last-applied-configuration of a resource to match the contents of a file; will create the annotation if it does not already exist

```
kubectl apply set-last-applied -f deploy.yaml --create-annotation=true
```

Set the latest last-applied-configuration annotations by setting it to match the contents of a file. This results in the last-applied-configuration being updated as though 'kubectl apply -f' was run, without updating any other parts of the object.

Usage

```
$ kubectl apply set-last-applied -f FILENAME
```

Flags

Name	Shorthand	Default	Usage
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 goolang and jsonpath output formats.
create-annotation		false	Will create 'last-applied-configuration' annotations if current objects doesn't have one
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.
filename	f	[]	Filename, directory, or URL to files that contains the last-applied-configuration annotations
output	o		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 goolang templates [http://golang.org/pkg/text/template/#pkg-overview].

view-last-applied

View the last-applied-configuration annotations by type/name in YAML

```
kubectl apply view-last-applied deployment/nginx
```

View the last-applied-configuration annotations by file in JSON

```
kubectl apply view-last-applied -f deploy.yaml -o json
```

View the latest last-applied-configuration annotations by type/name or file.

The default output will be printed to stdout in YAML format. You can use the -o option to change the output format.

Usage

\$ kubectl apply view-last-applied (TYPE [NAME | -l label] | TYPE/NAME | -f FILENAME)

Flags

Name	Shorthand	Default	Usage
all		false	Select all resources in the namespace of the specified resource types
filename	f	[]	Filename, directory, or URL to files that contains the last-applied-configuration annotations
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o	yaml	Output format. Must be one of yaml json
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
selector	l		Selector (label query) to filter on, supports '=', '==', and '!='.(e.g. -l key1=value1,key2=value2)

annotate

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-
```

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.

Usage

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

Flags

Name	Shorthand	Default	Usage
all		false	Select all resources, including uninitialized ones, in the namespace of the specified resource types.
all-namespaces	A	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 goyaml 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.
filename	f	[]	Filename, directory, or URL to files identifying the resource to update the annotation
kustomize	k		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.

Name	Shorthand	Default	Usage
local		false	If true, annotation will NOT contact api-server but run locally.
output	o		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.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	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.
selector	l		Selector (label query) to filter on, not including uninitialized ones, 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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].

autoscale

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
```

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.

Usage

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

Flags

Name	Shorthand	Default	Usage
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 goyaml 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.
filename	f	[]	Filename, directory, or URL to files identifying the resource to autoscale.
kustomize	k		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.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	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.
		false	If true, keep the managedFields when printing objects in JSON or YAML format.

Name	Shorthand	Default	Usage
show-managed-fields			
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is go lang templates [http://golang.org/pkg/text/template/#pkg-overview].

debug

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
```

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.

Usage

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

Flags

Name	Shorthand	Default	Usage
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.
container	c		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.
quiet	q	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.
stdin	i	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.
tty	t	false	Allocate a TTY for the debugging container.

diff

Diff resources included in pod.json

```
kubectl diff -f pod.json
```

Diff file read from stdin

```
cat service.yaml | kubectl diff -f -
```

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.

Usage

```
$ kubectl diff -f FILENAME
```

Flags

Name	Shorthand	Default	Usage
field-manager		kubectl-client-side-apply	Name of the manager used to track field ownership.
filename	f	[]	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.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
selector	l		Selector (label query) to filter on, supports '=', '==', and '!='.(e.g. -l key1=value1,key2=value2)
server-side		false	If true, apply runs in the server instead of the client.

edit

Edit the service named 'docker-registry'

```
kubectl edit svc/docker-registry
```

Use an alternative editor

```
KUBE_EDITOR="nano" kubectl edit svc/docker-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 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.

Usage

```
$ kubectl edit (RESOURCE/NAME | -f FILENAME)
```

Flags

Name	Shorthand	Default	Usage
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 goyang and jsonpath output formats.
field-manager		kubectl-edit	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files to use to edit the resource

Name	Shorthand	Default	Usage
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		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.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it
windows-line-endings		false	Defaults to the line ending native to your platform.

kustomize

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
```

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.

Usage

\$ kubectl kustomize DIR

Flags

Name	Shorthand	Default	Usage
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.
enable-managedby-label		false	enable adding app.kubernetes.io/managed-by
env	e	[]	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
output	o		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.

label

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-
```

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.

Usage

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

Flags

Name	Shorthand	Default	Usage
all		false	Select all resources, including uninitialized ones, in the namespace of the specified resource types
all-namespaces	A	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 go lang 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.

Name	Shorthand	Default	Usage
filename	f	[]	Filename, directory, or URL to files identifying the resource to update the labels
kustomize	k		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.
output	o		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.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	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.
selector	l		Selector (label query) to filter on, not including uninitialized ones, 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].

patch

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":"new image"}]}}'
```

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 fields of a resource using strategic merge patch, a JSON merge patch, or a JSON patch.

JSON and YAML formats are accepted.

Usage

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

Flags

Name	Shorthand	Default	Usage
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 goyaml 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.
filename	f	[]	Filename, directory, or URL to files identifying the resource to update
kustomize	k		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.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
patch	p		The patch to be applied to the resource JSON file.
patch-file			A file containing a patch to be applied to the resource.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.

Name	Shorthand	Default	Usage
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].
type		strategic	The type of patch being provided; one of [json merge strategic]

replace

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
```

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
```

Usage

```
$ kubectl replace -f FILENAME
```

Flags

Name	Shorthand	Default	Usage
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

Name	Shorthand	Default	Usage
			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.
filename	f	[]	to use to replace the resource.
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).
kustomize	k		Process a kustomization directory. This flag can't be used together with -f or -R.
output	o		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.
recursive	R	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].
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		true	If true, use a schema to validate the input before sending it
wait		false	If true, wait for resources to be gone before returning. This waits for finalizers.

rollout

Rollback to the previous deployment

```
kubectl rollout undo deployment/abc
```

Check the rollout status of a daemonset

```
kubectl rollout status daemonset/foo
```

Manage the rollout of a resource.

Valid resource types include:

- deployments
- daemonsets
- statefulsets

Usage

```
$ kubectl rollout SUBCOMMAND
```

history

View the rollout history of a deployment

```
kubectl rollout history deployment/abc
```

View the details of daemonset revision 3

```
kubectl rollout history daemonset/abc --revision=3
```

View previous rollout revisions and configurations.

Usage

```
$ kubectl rollout history (TYPE NAME | TYPE/NAME) [flags]
```

Flags

Name	Shorthand	Default	Usage
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.
filename	f	[]	Filename, directory, or URL to files identifying the resource to get from a server.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		

Name	Shorthand	Default	Usage
			Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
revision		0	See the details, including podTemplate of the revision specified
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].

pause

Mark the nginx deployment as paused # Any current state of the deployment will continue its function; new updates # to the deployment will not have an effect as long as the deployment is paused

```
kubectl rollout pause deployment/nginx
```

Mark the provided resource as paused.

Paused resources will not be reconciled by a controller. Use "kubectl rollout resume" to resume a paused resource. Currently only deployments support being paused.

Usage

```
$ kubectl rollout pause RESOURCE
```

Flags

Name	Shorthand	Default	Usage
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- rollout	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files identifying the resource to get from a server.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.

Name	Shorthand	Default	Usage
recursive	R	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.
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is go lang templates [http://golang.org/pkg/text/template/#pkg-overview].

restart

Restart a deployment

```
kubectl rollout restart deployment/nginx
```

Restart a daemon set

```
kubectl rollout restart daemonset/abc
```

Restart a resource.

Resource rollout will be restarted.

Usage

```
$ kubectl rollout restart RESOURCE
```

Flags

Name	Shorthand	Default	Usage
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 go lang and jsonpath output formats.
field-manager		kubectl-rollout	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files identifying the resource to get from a server.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
		false	If true, keep the managedFields when printing objects in JSON or YAML format.

Name	Shorthand	Default	Usage
show-managed-fields			
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is go lang templates [http://golang.org/pkg/text/template/#pkg-overview].

resume

Resume an already paused deployment

```
kubectl rollout resume deployment/nginx
```

Resume a paused resource.

Paused resources will not be reconciled by a controller. By resuming a resource, we allow it to be reconciled again. Currently only deployments support being resumed.

Usage

```
$ kubectl rollout resume RESOURCE
```

Flags

Name	Shorthand	Default	Usage
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 go lang and jsonpath output formats.
field-manager		kubectl-rollout	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files identifying the resource to get from a server.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
recursive	R	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.
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is go lang templates [http://golang.org/pkg/text/template/#pkg-overview].

status

Watch the rollout status of a deployment

```
kubectl rollout status deployment/nginx
```

Show the status of the rollout.

By default 'rollout status' will watch the status of the latest rollout until it's done. If you don't want to wait for the rollout to finish then you can use `--watch=false`. Note that if a new rollout starts in-between, then 'rollout status' will continue watching the latest revision. If you want to pin to a specific revision and abort if it is rolled over by another revision, use `--revision=N` where N is the revision you need to watch for.

Usage

```
$ kubectl rollout status (TYPE NAME | TYPE/NAME) [flags]
```

Flags

Name	Shorthand	Default	Usage
filename	f	[]	Filename, directory, or URL to files identifying the resource to get from a server.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
revision		0	Pin to a specific revision for showing its status. Defaults to 0 (last revision).
timeout		0s	The length of time to wait before ending watch, zero means never. Any other values should contain a corresponding time unit (e.g. 1s, 2m, 3h).
watch	w	true	Watch the status of the rollout until it's done.

undo

Roll back to the previous deployment

```
kubectl rollout undo deployment/abc
```

Roll back to daemonset revision 3

```
kubectl rollout undo daemonset/abc --to-revision=3
```

Roll back to the previous deployment with dry-run

```
kubectl rollout undo --dry-run=server deployment/abc
```

Roll back to a previous rollout.

Usage

\$ kubectl rollout undo (TYPE NAME | TYPE/NAME) [flags]

Flags

Name	Shorthand	Default	Usage
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.
filename	f	[]	Filename, directory, or URL to files identifying the resource to get from a server.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
recursive	R	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.
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] .
to-revision		0	The revision to rollback to. Default to 0 (last revision).

scale

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
```

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.

Usage

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

Flags

Name	Shorthand	Default	Usage
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 goyaml 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.
filename	f	[]	Filename, directory, or URL to files identifying the resource to set a new size
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	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.
selector	l		

Name	Shorthand	Default	Usage
			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 go lang 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).

set

Configure application resources.

These commands help you make changes to existing application resources.

Usage

```
$ kubectl set SUBCOMMAND
```

env

Update deployment 'registry' with a new environment variable

```
kubectl set env deployment/registry STORAGE_DIR=/local
```

List the environment variables defined on a deployments 'sample-build'

```
kubectl set env deployment/sample-build --list
```

List the environment variables defined on all pods

```
kubectl set env pods --all --list
```

Output modified deployment in YAML, and does not alter the object on the server

```
kubectl set env deployment/sample-build STORAGE_DIR=/data -o yaml
```

Update all containers in all replication controllers in the project to have ENV=prod

```
kubectl set env rc --all ENV=prod
```

Import environment from a secret

```
kubectl set env --from=secret/mysecret deployment/myapp
```

Import environment from a config map with a prefix

```
kubectl set env --from=configmap/myconfigmap --prefix=MYSQL_ deployment/myapp
```

Import specific keys from a config map

```
kubectl set env --keys=my-example-key --from=configmap/myconfigmap deployment/myapp
```

Remove the environment variable ENV from container 'c1' in all deployment configs

```
kubectl set env deployments --all --containers="c1" ENV-
```

Remove the environment variable ENV from a deployment definition on disk and # update the deployment config on the server

```
kubectl set env -f deploy.json ENV-
```

Set some of the local shell environment into a deployment config on the server

```
env | grep RAILS_ | kubectl set env -e - deployment/registry
```

Update environment variables on a pod template.

List environment variable definitions in one or more pods, pod templates. Add, update, or remove container environment variable definitions in one or more pod templates (within replication controllers or deployment configurations). View or modify the environment variable definitions on all containers in the specified pods or pod templates, or just those that match a wildcard.

If "--env -" is passed, environment variables can be read from STDIN using the standard env syntax.

Possible resources include (case insensitive):

pod (po), replicationcontroller (rc), deployment (deploy), daemonset (ds), statefulset (sts), cronjob (cj), replicaset (rs)

Usage

```
$ kubectl set env RESOURCE/NAME KEY_1=VAL_1 ... KEY_N=VAL_N
```

Flags

Name	Shorthand	Default	Usage
all		false	If true, 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 go lang and jsonpath output formats.
containers	c	*	The names of containers in the selected pod templates to change - may use wildcards
dry-run		none	

Name	Shorthand	Default	Usage
			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	e	[]	Specify a key-value pair for an environment variable to set into each container.
field-manager		kubectl-set	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files the resource to update the env
from			The name of a resource from which to inject environment variables
keys		[]	Comma-separated list of keys to import from specified resource
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
list		false	If true, display the environment and any changes in the standard format. this flag will removed when we have kubectl view env.
local		false	If true, set env will NOT contact api-server but run locally.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
overwrite		true	If true, allow environment to be overwritten, otherwise reject updates that overwrite existing environment.
prefix			Prefix to append to variable names
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
resolve		false	If true, show secret or configmap references when listing variables
selector	l		Selector (label query) to filter on
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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].

image

Set a deployment's nginx container image to 'nginx:1.9.1', and its busybox container image to 'busybox'

```
kubectl set image deployment/nginx busybox=busybox nginx=nginx:1.9.1
```

Update all deployments' and rc's nginx container's image to 'nginx:1.9.1'

```
kubectl set image deployments,rc nginx=nginx:1.9.1 --all
```

Update image of all containers of daemonset abc to 'nginx:1.9.1'

```
kubectl set image daemonset abc *=nginx:1.9.1
```

Print result (in yaml format) of updating nginx container image from local file, without hitting the server

```
kubectl set image -f path/to/file.yaml nginx=nginx:1.9.1 --local -o yaml
```

Update existing container image(s) of resources.

Possible resources include (case insensitive):

pod (po), replicationcontroller (rc), deployment (deploy), daemonset (ds), statefulset (sts), cronjob (cj), replicaset (rs)

Usage

```
$ kubectl set image (-f FILENAME | TYPE NAME)
CONTAINER_NAME_1=CONTAINER_IMAGE_1 ...
CONTAINER_NAME_N=CONTAINER_IMAGE_N
```

Flags

Name	Shorthand	Default	Usage
all		false	Select all resources, including uninitialized ones, 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 go lang 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-set	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files identifying the resource to get from a server.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
local		false	If true, set image will NOT contact api-server but run locally.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.

Name	Shorthand	Default	Usage
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
selector	l		Selector (label query) to filter on, not including uninitialized ones, 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 goolang templates [http://golang.org/pkg/text/template/#pkg-overview].

resources

Set a deployments nginx container cpu limits to "200m" and memory to "512Mi"

```
kubectl set resources deployment nginx -c=nginx --limits=cpu=200m,memory=512Mi
```

Set the resource request and limits for all containers in nginx

```
kubectl set resources deployment nginx --limits=cpu=200m,memory=512Mi --requests=cpu=100m,memory=256Mi
```

Remove the resource requests for resources on containers in nginx

```
kubectl set resources deployment nginx --limits=cpu=0,memory=0 --requests=cpu=0,memory=0
```

Print the result (in yaml format) of updating nginx container limits from a local, without hitting the server

```
kubectl set resources -f path/to/file.yaml --limits=cpu=200m,memory=512Mi --local -o yaml
```

Specify compute resource requirements (CPU, memory) for any resource that defines a pod template. If a pod is successfully scheduled, it is guaranteed the amount of resource requested, but may burst up to its specified limits.

For each compute resource, if a limit is specified and a request is omitted, the request will default to the limit.

Possible resources include (case insensitive): Use "kubectl api-resources" for a complete list of supported resources..

Usage

```
$ kubectl set resources (-f FILENAME | TYPE NAME) ([--limits=LIMITS & --requests=REQUESTS]
```

Flags

Name	Shorthand	Default	Usage
all		false	Select all resources, including uninitialized ones, 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 go lang and jsonpath output formats.
containers	c	*	The names of containers in the selected pod templates to change, all containers are selected by default - may use wildcards
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-set	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files identifying the resource to get from a server.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
limits			The resource requirement requests for this container. For example, 'cpu=100m,memory=256Mi'. Note that server side components may assign requests depending on the server configuration, such as limit ranges.
local		false	If true, set resources will NOT contact api-server but run locally.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
requests			The resource requirement requests for this container. For example, 'cpu=100m,memory=256Mi'. Note that server side components may assign requests depending on the server configuration, such as limit ranges.
selector	l		Selector (label query) to filter on, not including uninitialized ones, 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

Name	Shorthand	Default	Usage
			golang templates [http://golang.org/pkg/text/template/#pkg-overview].

selector

Set the labels and selector before creating a deployment/service pair

```
kubectl create service clusterip my-svc --clusterip="None" -o yaml --dry-run=client | kubectl set
selector --local -f - 'environment=qa' -o yaml | kubectl create -f -
kubectl create deployment my-dep -o yaml --dry-run=client | kubectl label --local -f - environm
ent=qa -o yaml | kubectl create -f -
```

Set the selector on a resource. Note that the new selector will overwrite the old selector if the resource had one prior to the invocation of 'set selector'.

A selector must begin with a letter or number, and may contain letters, numbers, hyphens, dots, and underscores, up to 63 characters. If --resource-version is specified, then updates will use this resource version, otherwise the existing resource-version will be used. Note: currently selectors can only be set on Service objects.

Usage

```
$ kubectl set selector (-f FILENAME | TYPE NAME) EXPRESSIONS [--resource-
version=version]
```

Flags

Name	Shorthand	Default	Usage
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 goyaml 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-set	Name of the manager used to track field ownership.
filename	f	[]	identifying the resource.
local		false	If true, annotation will NOT contact api-server but run locally.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set

Name	Shorthand	Default	Usage
			to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	true	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 selectors update will only succeed if this is the current resource-version for the object. Only valid when specifying a single resource.
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 goolang templates [http://golang.org/pkg/text/template/#pkg-overview].

serviceaccount

Set deployment nginx-deployment's service account to serviceaccount1

```
kubectl set serviceaccount deployment nginx-deployment serviceaccount1
```

Print the result (in YAML format) of updated nginx deployment with the service account from local file, without hitting the API server

```
kubectl set sa -f nginx-deployment.yaml serviceaccount1 --local --dry-run=client -o yaml
```

Update the service account of pod template resources.

Possible resources (case insensitive) can be:

replicationcontroller (rc), deployment (deploy), daemonset (ds), job, replicaset (rs), statefulset

Usage

\$ kubectl set serviceaccount (-f FILENAME | TYPE NAME) SERVICE_ACCOUNT

Flags

Name	Shorthand	Default	Usage
all		false	Select all resources, including uninitialized ones, 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 goolang 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.

Name	Shorthand	Default	Usage
field-manager		kubectl-set	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files identifying the resource to get from a server.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
local		false	If true, set serviceaccount will NOT contact api-server but run locally.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
record		false	Record current kubectl command in the resource annotation. If set to false, do not record the command. If set to true, record the command. If not set, default to updating the existing annotation value only if one already exists.
recursive	R	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.
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is go lang templates [http://golang.org/pkg/text/template/#pkg-overview].

subject

Update a cluster role binding for serviceaccount1

```
kubectl set subject clusterrolebinding admin --serviceaccount=namespace:serviceaccount1
```

Update a role binding for user1, user2, and group1

```
kubectl set subject rolebinding admin --user=user1 --user=user2 --group=group1
```

Print the result (in YAML format) of updating rolebinding subjects from a local, without hitting the server

```
kubectl create rolebinding admin --role=admin --user=admin -o yaml --dry-run=client | kubectl set subject --local -f - --user=foo -o yaml
```

Update the user, group, or service account in a role binding or cluster role binding.

Usage

```
$ kubectl set subject (-f FILENAME | TYPE NAME) [--user=username] [--group=groupname] [--serviceaccount=namespace:serviceaccountname] [--dry-run=server|client|none]
```

Flags

Name	Shorthand	Default	Usage
all		false	Select all resources, including uninitialized ones, 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.
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-set	Name of the manager used to track field ownership.
filename	f	[]	Filename, directory, or URL to files the resource to update the subjects
group		[]	Groups to bind to the role
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
local		false	If true, set subject will NOT contact api-server but run locally.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
selector	l		Selector (label query) to filter on, not including uninitialized ones, supports '=', '==', and '!='.(e.g. -l key1=value1,key2=value2)
serviceaccount		[]	Service accounts to bind to the role
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].

wait

Wait for the pod "busybox1" to contain the status condition of type "Ready"

```
kubectrl wait --for=condition=Ready pod/busybox1
```

The default value of status condition is true; you can set it to false

```
kubectrl wait --for=condition=Ready=false 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
```

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.

Usage

```
$ kubectl wait ([-f FILENAME] | resource.group/resource.name | resource.group [-l label | --all]) [--for=delete|--for condition=available]
```

Flags

Name	Shorthand	Default	Usage
all		false	Select all resources in the namespace of the specified resource types
all-namespaces	A	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.
filename	f	[]	identifying the resource.
for			The condition to wait on: [delete condition=condition-name]. The default status value of condition-name is true, you can set false with condition=condition-name=false
local		false	If true, annotation will NOT contact api-server but run locally.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
recursive	R	true	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
selector	l		Selector (label query) to filter on, supports '=', '==', and '!='. (e.g. -l key1=value1,key2=value2)
		false	

Name	Shorthand	Default	Usage
show-managed-fields			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.

WORKING WITH APPS

This section contains commands for inspecting and debugging your applications.

- logs will print the logs from the specified pod + container.
- exec can be used to get an interactive shell on a pod + container.
- describe will print debug information about the given resource.

attach

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
```

Attach to a process that is already running inside an existing container.

Usage

```
$ kubectl attach (POD | TYPE/NAME) -c CONTAINER
```

Flags

Name	Shorthand	Default	Usage
container	c		Container name. If omitted, use the <code>kubectl.kubernetes.io/default-container</code> 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
quiet	q	false	Only print output from the remote session
stdin	i	false	Pass stdin to the container
tty	t	false	Stdin is a TTY

auth

Inspect authorization

Usage

\$ kubectl auth

can-i

Check to see if I can create pods in any namespace

```
kubectl auth can-i create pods --all-namespaces
```

Check to see if I can list deployments in my current namespace

```
kubectl auth can-i list deployments.apps
```

Check to see if I can do everything in my current namespace ("*" means all)

```
kubectl auth can-i '*' '*'
```

Check to see if I can get the job named "bar" in namespace "foo"

```
kubectl auth can-i list jobs.batch/bar -n foo
```

Check to see if I can read pod logs

```
kubectl auth can-i get pods --subresource=log
```

Check to see if I can access the URL /logs/

```
kubectl auth can-i get /logs/
```

List all allowed actions in namespace "foo"

```
kubectl auth can-i --list --namespace=foo
```

Check whether an action is allowed.

VERB is a logical Kubernetes API verb like 'get', 'list', 'watch', 'delete', etc. TYPE is a Kubernetes resource. Shortcuts and groups will be resolved. NONRESOURCEURL is a partial URL that starts with "/". NAME is the name of a particular Kubernetes resource.

Usage

```
$ kubectl auth can-i VERB [TYPE | TYPE/NAME | NONRESOURCEURL]
```

Flags

Name	Shorthand	Default	Usage
all-namespaces	A	false	If true, check the specified action in all namespaces.
list		false	If true, prints all allowed actions.
no-headers		false	If true, prints allowed actions without headers
quiet	q	false	If true, suppress output and just return the exit code.
subresource			SubResource such as pod/log or deployment/scale

reconcile

Reconcile RBAC resources from a file

```
kubectl auth reconcile -f my-rbac-rules.yaml
```

Reconciles rules for RBAC role, role binding, cluster role, and cluster role binding objects.

Missing objects are created, and the containing namespace is created for namespaced objects, if required.

Existing roles are updated to include the permissions in the input objects, and remove extra permissions if --remove-extra-permissions is specified.

Existing bindings are updated to include the subjects in the input objects, and remove extra subjects if --remove-extra-subjects is specified.

This is preferred to 'apply' for RBAC resources so that semantically-aware merging of rules and subjects is done.

Usage

```
$ kubectl auth reconcile -f FILENAME
```

Flags

Name	Shorthand	Default	Usage
allow-missing-template-keys		true	

Name	Shorthand	Default	Usage
			If true, ignore any errors in templates when a field or map key is missing in the template. Only applies to goyaml 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.
filename	f	[]	Filename, directory, or URL to files identifying the resource to reconcile.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
remove-extra-permissions		false	If true, removes extra permissions added to roles
remove-extra-subjects		false	If true, removes extra subjects added to rolebindings
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 goyaml templates [http://golang.org/pkg/text/template/#pkg-overview].

cp

!!!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

```
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

```
kubectrl cp /tmp/foo <some-namespace>/<some-pod>:/tmp/bar
```

Copy /tmp/foo from a remote pod to /tmp/bar locally

```
kubectrl cp <some-namespace>/<some-pod>:/tmp/foo /tmp/bar
```

Copy files and directories to and from containers.

Usage

```
$ kubectrl cp <file-spec-src> <file-spec-dest>
```

Flags

Name	Shorthand	Default	Usage
container	c		Container name. If omitted, use the kubectrl.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

describe

Describe a node

```
kubectrl describe nodes kubernetes-node-emt8.c.myproject.internal
```

Describe a pod

```
kubectrl describe pods/nginx
```

Describe a pod identified by type and name in "pod.json"

```
kubectrl describe -f pod.json
```

Describe all pods

```
kubectrl describe pods
```

Describe pods by label name=myLabel

```
kubectrl 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 the name)

```
kubectrl describe pods frontend
```

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.

Usage

```
$ kubectl describe (-f FILENAME | TYPE [NAME_PREFIX | -l label] | TYPE/NAME)
```

Flags

Name	Shorthand	Default	Usage
all-namespaces	A	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.
filename	f	[]	Filename, directory, or URL to files containing the resource to describe
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
recursive	R	false	Process the directory used in -f, --filename recursively. Useful when you want to manage related manifests organized within the same directory.
selector	l		Selector (label query) to filter on, supports '=', '==', and '!='. (e.g. -l key1=value1,key2=value2)
show-events		true	If true, display events related to the described object.

exec

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
```

Execute a command in a container.

Usage

```
$ kubectl exec (POD | TYPE/NAME) [-c CONTAINER] [flags] -- COMMAND [args...]
```

Flags

Name	Shorthand	Default	Usage
container	c		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
filename	f	[]	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
quiet	q	false	Only print output from the remote session
stdin	i	false	Pass stdin to the container
tty	t	false	Stdin is a TTY

logs

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

```
kubectrl logs -p -c ruby web-1
```

Begin streaming the logs of the ruby container in pod web-1

```
kubectrl logs -f -c ruby web-1
```

Begin streaming the logs from all containers in pods defined by label app=nginx

```
kubectrl logs -f -l app=nginx --all-containers=true
```

Display only the most recent 20 lines of output in pod nginx

```
kubectrl logs --tail=20 nginx
```

Show all logs from pod nginx written in the last hour

```
kubectrl logs --since=1h nginx
```

Show logs from a kubelet with an expired serving certificate

```
kubectrl logs --insecure-skip-tls-verify-backend nginx
```

Return snapshot logs from first container of a job named hello

```
kubectrl logs job/hello
```

Return snapshot logs from container nginx-1 of a deployment named nginx

```
kubectrl logs deployment/nginx -c nginx-1
```

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

Usage

```
$ kubectrl logs [-f] [-p] (POD | TYPE/NAME) [-c CONTAINER]
```

Flags

Name	Shorthand	Default	Usage
all-containers		false	Get all containers' logs in the pod(s).
container	c		Print the logs of this container
follow	f	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.
		5	

Name	Shorthand	Default	Usage
max-log-requests			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)
previous	p	false	If true, print the logs for the previous instance of the container in a pod if it exists.
selector	l		Selector (label query) to filter on.
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

port-forward

Listen on ports 5000 and 6000 locally, forwarding data to/from ports 5000 and 6000 in the pod

```
kubectrl 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

```
kubectrl 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

```
kubectrl port-forward service/myservice 8443:https
```

Listen on port 8888 locally, forwarding to 5000 in the pod

```
kubectrl port-forward pod/mypod 8888:5000
```

Listen on port 8888 on all addresses, forwarding to 5000 in the pod

```
kubectrl 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

```
kubectrl port-forward --address localhost,10.19.21.23 pod/mypod 8888:5000
```

Listen on a random port locally, forwarding to 5000 in the pod

```
kubectrl port-forward pod/mypod :5000
```

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.

Usage

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

Flags

Name	Shorthand	Default	Usage
address		[localhost]	Addresses to listen on (comma separated). Only accepts IP addresses or localhost as a value. When localhost is supplied, kubectrl 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

proxy

To proxy all of the Kubernetes API and nothing else

```
kubectrl 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'

```
kubectrl 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'

```
kubectrl proxy --api-prefix=/custom/
```

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

```
kubectrl 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

```
kubectrl 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/

```
kubectrl proxy --api-prefix=/k8s-api
```

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.

Usage

```
$ kubectrl proxy [--port=PORT] [--www=static-dir] [--www-prefix=prefix] [--api-prefix=prefix]
```

Flags

Name	Shorthand	Default	Usage
accept-hosts		<code>^localhost\$,^127.0.0.1\$,^[:1]\$\</code>	Regular expression for hosts that the proxy should accept.
accept-paths		<code>^.*\</code>	Regular expression for paths that the proxy should accept.
address		<code>127.0.0.1</code>	The IP address on which to serve on.
api-prefix		<code>/</code>	Prefix to serve the proxied API under.
disable-filter		<code>false</code>	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		<code>0s</code>	keepalive specifies the keep-alive period for an active network connection. Set to 0 to disable keepalive.
port	<code>p</code>	<code>8001</code>	The port on which to run the proxy. Set to 0 to pick a random port.
reject-methods		<code>^\$\</code>	Regular expression for HTTP methods that the proxy should reject (example <code>--reject-methods='POST,PUT,PATCH'</code>).
reject-paths		<code>^/api/./pods/./exec,^/api/./pods/./attach\</code>	Regular expression for paths that the proxy should reject. Paths specified here will be rejected even accepted by <code>--accept-paths</code> .
unix-socket	<code>u</code>		Unix socket on which to run the proxy.
www	<code>w</code>		Also serve static files from the given directory under the specified prefix.
www-prefix	<code>P</code>	<code>/static/</code>	Prefix to serve static files under, if static file directory is specified.

top

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.

Usage

```
$ kubectl top
```

node

Show metrics for all nodes

```
kubectl top node
```

Show metrics for a given node

```
kubectl top node NODE_NAME
```

Display resource (CPU/memory) usage of nodes.

The top-node command allows you to see the resource consumption of nodes.

Usage

```
$ kubectl top node [NAME | -l label]
```

Flags

Name	Shorthand	Default	Usage
no-headers		false	If present, print output without headers
selector	<code>-l</code>		Selector (label query) to filter on, supports '=', '==', and '!='. (e.g. <code>-l key1=value1,key2=value2</code>)
sort-by			If non-empty, sort nodes list using specified field. The field can be either 'cpu' or 'memory'.
use-protocol- buffers		true	Enables using protocol-buffers to access Metrics API.

pod

Show metrics for all pods in the default namespace

```
kubectl top pod
```

Show metrics for all pods in the given namespace

```
kubectl top pod --namespace=NAMESPACE
```

Show metrics for a given pod and its containers

```
kubectl top pod POD_NAME --containers
```

Show metrics for the pods defined by label name=myLabel

```
kubectl top pod -l name=myLabel
```

Display resource (CPU/memory) usage of pods.

The 'top pod' command allows you to see the resource consumption of pods.

Due to the metrics pipeline delay, they may be unavailable for a few minutes since pod creation.

Usage

```
$ kubectl top pod [NAME | -l label]
```

Flags

Name	Shorthand	Default	Usage
all-namespaces	A	false	If present, list the requested object(s) across all namespaces. Namespace in current context is ignored even if specified with --namespace.
containers		false	If present, print usage of containers within a pod.
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.
no-headers		false	If present, print output without headers.
selector	l		Selector (label query) to filter on, supports '=', '==', and '!='. (e.g. -l key1=value1,key2=value2)
sort-by			If non-empty, sort pods list using specified field. The field can be either 'cpu' or 'memory'.
use-protocol-buffers		true	Enables using protocol-buffers to access Metrics API.

CLUSTER MANAGEMENT

api-versions

Print the supported API versions

```
kubectl api-versions
```

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

Usage

\$ kubectl api-versions

certificate

Modify certificate resources.

Usage

\$ kubectl certificate SUBCOMMAND

approve

Approve CSR 'csr-sqgzp'

kubectl certificate approve csr-sqgzp

Approve a certificate signing request.

kubectl certificate approve allows a cluster admin to approve a certificate signing request (CSR). This action tells a certificate signing controller to issue a certificate to the requestor with the attributes requested in the CSR.

SECURITY NOTICE: Depending on the requested attributes, the issued certificate can potentially grant a requester access to cluster resources or to authenticate as a requested identity. Before approving a CSR, ensure you understand what the signed certificate can do.

Usage

\$ kubectl certificate approve (-f FILENAME | NAME)

Flags

Name	Shorthand	Default	Usage
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.
filename	f	[]	Filename, directory, or URL to files identifying the resource to update
force		false	Update the CSR even if it is already approved.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
recursive	R	false	

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

deny

Deny CSR 'csr-sqgzp'

```
kubectl certificate deny csr-sqgzp
```

Deny a certificate signing request.

kubectl certificate deny allows a cluster admin to deny a certificate signing request (CSR). This action tells a certificate signing controller to not to issue a certificate to the requestor.

Usage

\$ kubectl certificate deny (-f FILENAME | NAME)

Flags

Name	Shorthand	Default	Usage
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 goolang and jsonpath output formats.
filename	f	[]	Filename, directory, or URL to files identifying the resource to update
force		false	Update the CSR even if it is already denied.
kustomize	k		Process the kustomization directory. This flag can't be used together with -f or -R.
output	o		Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
recursive	R	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.
template			Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format is

Name	Shorthand	Default	Usage
			golang templates [http://golang.org/pkg/text/template/#pkg-overview].

cluster-info

Print the address of the control plane and cluster services

```
kubectl cluster-info
```

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'`.

Usage

```
$ kubectl cluster-info
```

dump

Dump current cluster state to stdout

```
kubectl cluster-info dump
```

Dump current cluster state to `/path/to/cluster-state`

```
kubectl cluster-info dump --output-directory=/path/to/cluster-state
```

Dump all namespaces to stdout

```
kubectl cluster-info dump --all-namespaces
```

Dump a set of namespaces to `/path/to/cluster-state`

```
kubectl cluster-info dump --namespaces default,kube-system --output-directory=/path/to/cluster-state
```

Dump cluster information out suitable for debugging and diagnosing cluster problems. By default, dumps everything to stdout. You can optionally specify a directory with `--output-directory`. If you specify a directory, Kubernetes will build a set of files in that directory. By default, only dumps things in the current namespace and 'kube-system' namespace, but you can switch to a different namespace with the `--namespaces` flag, or specify `--all-namespaces` to dump all namespaces.

The command also dumps the logs of all of the pods in the cluster; these logs are dumped into different directories based on namespace and pod name.

Usage

```
$ kubectl cluster-info dump
```

Flags

Name	Shorthand	Default	Usage
all-namespaces	A	false	If true, dump all namespaces. If true, --namespaces is ignored.
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.
namespaces		[]	A comma separated list of namespaces to dump.
output	o	json	Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
output-directory			Where to output the files. If empty or '-' uses stdout, otherwise creates a directory hierarchy in that directory
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
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].

cordon

Mark node "foo" as unschedulable

```
kubectcl cordon foo
```

Mark node as unschedulable.

Usage

```
$ kubectcl cordon NODE
```

Flags

Name	Shorthand	Default	Usage
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.
selector	l		Selector (label query) to filter on

drain

Drain node "foo", even if there are pods not managed by a replication controller, replica set, job, daemon set or stateful set on it

```
kubectldrain 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

```
kubectldrain foo --grace-period=900
```

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/> . 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/kubectldrain.svg>

Usage

```
$ kubectldrain NODE
```

Flags

Name	Shorthand	Default	Usage
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).
delete-local-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	

Name	Shorthand	Default	Usage
			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 not managed by a ReplicationController, ReplicaSet, Job, DaemonSet or StatefulSet.
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.
ignore-errors		false	Ignore errors occurred between drain nodes in group.
pod-selector			Label selector to filter pods on the node
selector	l		Selector (label query) to filter on
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

taint

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
```

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.

Usage

```
$ kubectl taint NODE NAME KEY_1=VAL_1:TAINT_EFFECT_1 ...
KEY_N=VAL_N:TAINT_EFFECT_N
```

Flags

Name	Shorthand	Default	Usage
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 go lang 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.
output	o		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.
selector	l		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 go lang templates [http://golang.org/pkg/text/template/#pkg-overview].
validate		true	If true, use a schema to validate the input before sending it

uncordon

Mark node "foo" as schedulable

```
kubectl uncordon foo
```

Mark node as schedulable.

Usage

\$ kubectl uncordon NODE

Flags

Name	Shorthand	Default	Usage
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.
selector	l		Selector (label query) to filter on

KUBECTL SETTINGS AND USAGE

alpha

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

Usage

\$ kubectl alpha

api-resources

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=extensions
```

Print the supported API resources on the server.

Usage

```
$ kubectl api-resources
```

Flags

Name	Shorthand	Default	Usage
api-group			Limit to resources in the specified API group.
cached		false	Use the cached list of resources if available.
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).
output	o		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.

completion

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"
```

Output shell completion code for the specified shell (bash or zsh). 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.

Usage

```
$ kubectl completion SHELL
```

config

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.

Usage

```
$ kubectl config SUBCOMMAND
```

current-context

Display the current-context

```
kubectl config current-context
```

Display the current-context.

Usage

```
$ kubectl config current-context
```

delete-cluster

Delete the minikube cluster

```
kubectl config delete-cluster minikube
```

Delete the specified cluster from the kubeconfig.

Usage

```
$ kubectl config delete-cluster NAME
```

delete-context

Delete the context for the minikube cluster

```
kubectl config delete-context minikube
```

Delete the specified context from the kubeconfig.

Usage

```
$ kubectl config delete-context NAME
```

delete-user

Delete the minikube user

```
kubectl config delete-user minikube
```

Delete the specified user from the kubeconfig.

Usage

\$ kubectl config delete-user NAME

get-clusters

List the clusters that kubectl knows about

kubectl config get-clusters

Display clusters defined in the kubeconfig.

Usage

\$ kubectl config get-clusters

get-contexts

List all the contexts in your kubeconfig file

kubectl config get-contexts

Describe one context in your kubeconfig file

kubectl config get-contexts my-context

Display one or many contexts from the kubeconfig file.

Usage

\$ kubectl config get-contexts [(-o|--output=)name)]

Flags

Name	Shorthand	Default	Usage
no-headers		false	When using the default or custom-column output format, don't print headers (default print headers).
output	o		Output format. One of: name

get-users

List the users that kubectl knows about

kubectl config get-users

Display users defined in the kubeconfig.

Usage

```
$ kubectl config get-users
```

rename-context

Rename the context 'old-name' to 'new-name' in your kubeconfig file

```
kubectl config rename-context old-name new-name
```

Renames a context from the kubeconfig file.

CONTEXT_NAME is the context name that you want to change.

NEW_NAME is the new name you want to set.

Note: If the context being renamed is the 'current-context', this field will also be updated.

Usage

```
$ kubectl config rename-context CONTEXT_NAME NEW_NAME
```

set

Set the server field on the my-cluster cluster to <https://1.2.3.4>

```
kubectl config set clusters.my-cluster.server https://1.2.3.4
```

Set the certificate-authority-data field on the my-cluster cluster

```
kubectl config set clusters.my-cluster.certificate-authority-data $(echo "cert_data_here" | base64 -i -)
```

Set the cluster field in the my-context context to my-cluster

```
kubectl config set contexts.my-context.cluster my-cluster
```

Set the client-key-data field in the cluster-admin user using --set-raw-bytes option

```
kubectl config set users.cluster-admin.client-key-data cert_data_here --set-raw-bytes=true
```

Set an individual value in a kubeconfig file.

PROPERTY_NAME is a dot delimited name where each token represents either an attribute name or a map key. Map keys may not contain dots.

PROPERTY_VALUE is the new value you want to set. Binary fields such as 'certificate-authority-data' expect a base64 encoded string unless the --set-raw-bytes flag is used.

Specifying an attribute name that already exists will merge new fields on top of existing values.

Usage

```
$ kubectl config set PROPERTY_NAME PROPERTY_VALUE
```

Flags

Name	Shorthand	Default	Usage
set-raw-bytes		false	When writing a []byte PROPERTY_VALUE, write the given string directly without base64 decoding.

set-cluster

Set only the server field on the e2e cluster entry without touching other values

```
kubectl config set-cluster e2e --server=https://1.2.3.4
```

Embed certificate authority data for the e2e cluster entry

```
kubectl config set-cluster e2e --embed-certs --certificate-authority=~/.kube/e2e/
kubernetes.ca.crt
```

Disable cert checking for the dev cluster entry

```
kubectl config set-cluster e2e --insecure-skip-tls-verify=true
```

Set custom TLS server name to use for validation for the e2e cluster entry

```
kubectl config set-cluster e2e --tls-server-name=my-cluster-name
```

Set a cluster entry in kubeconfig.

Specifying a name that already exists will merge new fields on top of existing values for those fields.

Usage

```
$ kubectl config set-cluster NAME [--server=server] [--certificate-authority=path/to/certificate/
authority] [--insecure-skip-tls-verify=true] [--tls-server-name=example.com]
```

Flags

Name	Shorthand	Default	Usage
embed-certs		false	embed-certs for the cluster entry in kubeconfig

set-context

Set the user field on the gce context entry without touching other values

```
kubectl config set-context gce --user=cluster-admin
```


Set a context entry in kubeconfig.

Specifying a name that already exists will merge new fields on top of existing values for those fields.

Usage

```
$ kubectl config set-context [NAME | --current] [--cluster=cluster_nickname] [--user=user_nickname] [--namespace=namespace]
```

Flags

Name	Shorthand	Default	Usage
current		false	Modify the current context

set-credentials

Set only the "client-key" field on the "cluster-admin" # entry, without touching other values

```
kubectl config set-credentials cluster-admin --client-key=~/.kube/admin.key
```

Set basic auth for the "cluster-admin" entry

```
kubectl config set-credentials cluster-admin --username=admin --password=uXFGweU9l35qcif
```

Embed client certificate data in the "cluster-admin" entry

```
kubectl config set-credentials cluster-admin --client-certificate=~/.kube/admin.crt --embed-certs=true
```

Enable the Google Compute Platform auth provider for the "cluster-admin" entry

```
kubectl config set-credentials cluster-admin --auth-provider=gcp
```

Enable the OpenID Connect auth provider for the "cluster-admin" entry with additional args

```
kubectl config set-credentials cluster-admin --auth-provider=oidc --auth-provider-arg=client-id=foo --auth-provider-arg=client-secret=bar
```

Remove the "client-secret" config value for the OpenID Connect auth provider for the "cluster-admin" entry

```
kubectl config set-credentials cluster-admin --auth-provider=oidc --auth-provider-arg=client-secret-
```

Enable new exec auth plugin for the "cluster-admin" entry

```
kubectl config set-credentials cluster-admin --exec-command=/path/to/the/executable --exec-api-version=client.authentication.k8s.io/v1beta1
```

Define new exec auth plugin args for the "cluster-admin" entry

```
kubectcl config set-credentials cluster-admin --exec-arg=arg1 --exec-arg=arg2
```

Create or update exec auth plugin environment variables for the "cluster-admin" entry

```
kubectcl config set-credentials cluster-admin --exec-env=key1=val1 --exec-env=key2=val2
```

Remove exec auth plugin environment variables for the "cluster-admin" entry

```
kubectcl config set-credentials cluster-admin --exec-env=var-to-remove-
```

Set a user entry in kubeconfig.

Specifying a name that already exists will merge new fields on top of existing values.

Client-certificate flags: --client-certificate=certfile --client-key=keyfile

Bearer token flags: --token=bearer_token

Basic auth flags: --username=basic_user --password=basic_password

Bearer token and basic auth are mutually exclusive.

Usage

```
$ kubectcl config set-credentials NAME [--client-certificate=path/to/certfile] [--client-key=path/to/keyfile] [--token=bearer_token] [--username=basic_user] [--password=basic_password] [--auth-provider=provider_name] [--auth-provider-arg=key=value] [--exec-command=exec_command] [--exec-api-version=exec_api_version] [--exec-arg=arg] [--exec-env=key=value]
```

Flags

Name	Shorthand	Default	Usage
auth-provider			Auth provider for the user entry in kubeconfig
auth-provider-arg		[]	'key=value' arguments for the auth provider
embed-certs		false	Embed client cert/key for the user entry in kubeconfig
exec-api-version			API version of the exec credential plugin for the user entry in kubeconfig
exec-arg		[]	New arguments for the exec credential plugin command for the user entry in kubeconfig
exec-command			Command for the exec credential plugin for the user entry in kubeconfig
exec-env		[]	'key=value' environment values for the exec credential plugin

unset

Unset the current-context

```
kubectrl config unset current-context
```

Unset namespace in foo context

```
kubectrl config unset contexts.foo.namespace
```

Unset an individual value in a kubeconfig file.

PROPERTY_NAME is a dot delimited name where each token represents either an attribute name or a map key. Map keys may not contain dots.

Usage

```
$ kubectrl config unset PROPERTY_NAME
```

use-context

Use the context for the minikube cluster

```
kubectrl config use-context minikube
```

Set the current-context in a kubeconfig file.

Usage

```
$ kubectrl config use-context CONTEXT_NAME
```

view

Show merged kubeconfig settings

```
kubectrl config view
```

Show merged kubeconfig settings and raw certificate data

```
kubectrl config view --raw
```

Get the password for the e2e user

```
kubectrl config view -o jsonpath='{.users[?(@.name == "e2e")].user.password}'
```

Display merged kubeconfig settings or a specified kubeconfig file.

You can use --output jsonpath={...} to extract specific values using a jsonpath expression.

Usage

```
$ kubectrl config view
```

Flags

Name	Shorthand	Default	Usage
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.
flatten		false	Flatten the resulting kubeconfig file into self-contained output (useful for creating portable kubeconfig files)
merge		true	Merge the full hierarchy of kubeconfig files
minify		false	Remove all information not used by current-context from the output
output	o	yaml	Output format. One of: json yaml name go-template go-template-file template templatefile jsonpath jsonpath-as-json jsonpath-file.
raw		false	Display raw byte data
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] .

explain

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
```

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.

Usage

```
$ kubectl explain RESOURCE
```

Flags

Name	Shorthand	Default	Usage
api-version			Get different explanations for particular API version (API group/version)
recursive		false	Print the fields of fields (Currently only 1 level deep)

options

Print flags inherited by all commands

```
kubectl options
```

Print the list of flags inherited by all commands

Usage

```
$ kubectl options
```

plugin

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/>

Usage

```
$ kubectl plugin [flags]
```

list

List all available plugin files on a user's PATH.

Available plugin files are those that are: - executable - anywhere on the user's PATH - begin with "kubectl-"

Usage

```
$ kubectl plugin list
```

Flags

Name	Shorthand	Default	Usage
name-only		false	If true, display only the binary name of each plugin, rather than its full path

version

Print the client and server versions for the current context

```
kubectl version
```

Print the client and server version information for the current context.

Usage

```
$ kubectl version
```

Flags

Name	Shorthand	Default	Usage
client		false	If true, shows client version only (no server required).
output	o		One of 'yaml' or 'json'.
short		false	If true, print just the version number.