API:

The REST API is the fundamental fabric of Kubernetes.

All operations and communications between components, and external user commands are REST API calls that the API Server handles. Consequently, everything in the Kubernetes platform is treated as an API object and has a corresponding entry in the [API](https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.22/).

API versioning

* Alpha:
  + The version names contain alpha (for example, v1alpha1).
  + The software may contain bugs. Enabling a feature may expose bugs. A feature may be disabled by default.
  + The support for a feature may be dropped at any time without notice.
  + The API may change in incompatible ways in a later software release without notice.
* Beta:
  + The version names contain beta (for example, v2beta3).
  + The software is well tested. Enabling a feature is considered safe. Features are enabled by default.
  + The support for a feature will not be dropped, though the details may change.
  + The software is not recommended for production uses. Subsequent releases may introduce incompatible changes. If you have multiple clusters which can be upgraded independently, you may be able to relax this restriction.
* Stable:
  + The version name is vX where X is an integer.
  + The stable versions of features appear in released software for many subsequent versions.

| **Example** | **Track** |
| --- | --- |
| v1 | GA (generally available, stable) |
| v1beta1 | Beta (pre-release) |
| v1alpha1 | Alpha (experimental) |

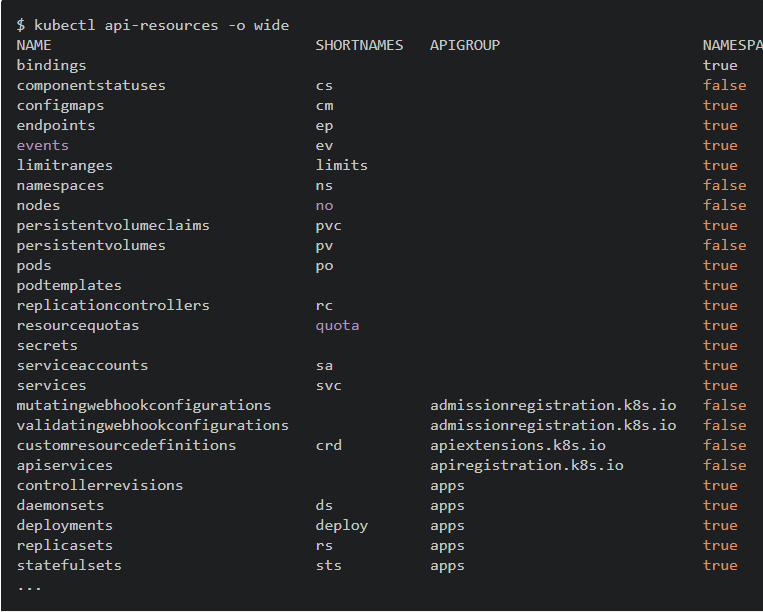
Enabling or disabling API groups

Certain resources and API groups are enabled by default. You can enable or disable them by setting --runtime-config on the API server.

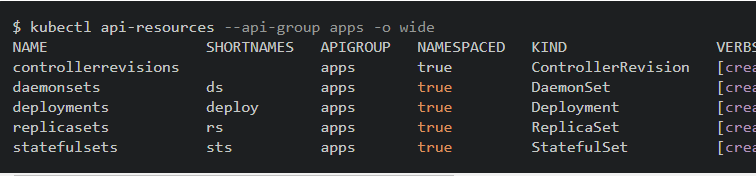
* to disable batch/v1, set --runtime-config=batch/v1=false
* to enable batch/v2alpha1, set --runtime-config=batch/v2alpha1

**Note:** When you enable or disable groups or resources, you need to restart the API server and controller manager to pick up the --runtime-config changes.

Kubectl api-resouces

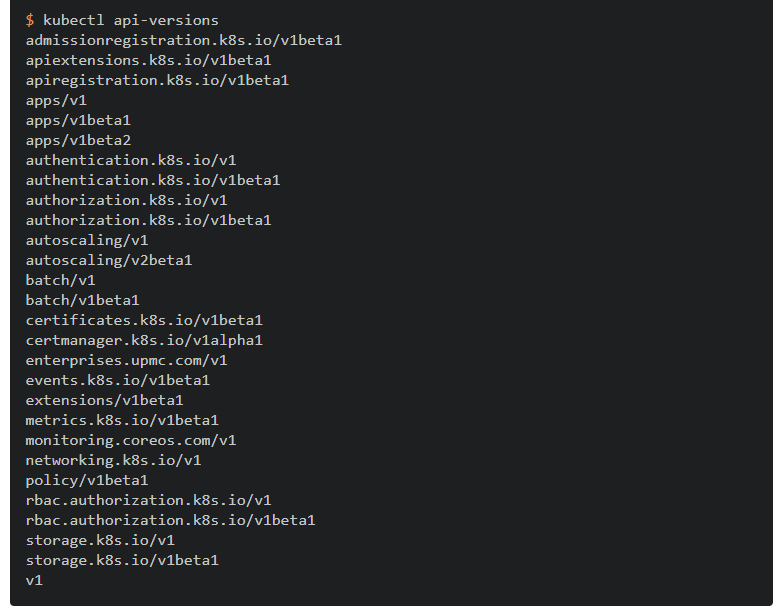


You also have the option to get API resources for a particular API group, for example:



### API Versions

You can also get all API versions supported by your cluster using this command:



## API endpoints for health

The Kubernetes API server provides 3 API endpoints (healthz, livez and readyz) to indicate the current status of the API server.

Machines that check the healthz/livez/readyz of the API server should rely on the HTTP status code. A status code 200 indicates the API server is healthy/live/ready, depending of the called endpoint

For all endpoints you can use the verbose parameter to print out the checks and their status. This can be useful for a human operator to debug the current status of the API server

curl -k https://localhost:6443/livez?verbose

or from a remote host with authentication:

kubectl get --raw='/readyz?verbose'

[+]ping ok

[+]log ok

[+]etcd ok

[+]poststarthook/start-kube-apiserver-admission-initializer ok

[+]poststarthook/generic-apiserver-start-informers ok

[+]poststarthook/start-apiextensions-informers ok

[+]poststarthook/start-apiextensions-controllers ok

[+]poststarthook/crd-informer-synced ok

[+]poststarthook/bootstrap-controller ok

[+]poststarthook/rbac/bootstrap-roles ok

[+]poststarthook/scheduling/bootstrap-system-priority-classes ok

[+]poststarthook/start-cluster-authentication-info-controller ok

[+]poststarthook/start-kube-aggregator-informers ok

[+]poststarthook/apiservice-registration-controller ok

[+]poststarthook/apiservice-status-available-controller ok

[+]poststarthook/kube-apiserver-autoregistration ok

[+]autoregister-completion ok

[+]poststarthook/apiservice-openapi-controller ok

healthz check passed