

### Viewing / Accessing resources

<code>kubectl get --all-namespaces {resource_type}</code>	Get resource from all namespaces
<code>kubectl describe {resource_type}</code>	Describes the given resources
<code>kubectl rolling-update {pod_name}</code>	Performs a rolling update for a given pod
<code>kubectl scale --replicas=3 {resource_type}/{resource_name}</code>	Scale given resource
<code>kubectl proxy</code>	Proxy the api server API to your local machine on port 8001
<code>kubectl port-forward {pod_name} {local_port}:{remote_port}</code>	Forward the pod port to your local port

### Creating /editing objects

<code>kubectl apply -f ./my-manifest.yaml</code>	Apply a configuration to a resource by filename or stdin. Also overrides the existing configuration.
<code>kubectl create -f ./my-manifest.yaml</code>	Create resource(s)
<code>kubectl create -f ./dir</code>	Create resource(s) in all manifest files in dir
<code>kubectl create -f https://git.io/vPieo`</code>	Create from url
<code>kubectl run nginx --image=nginx</code>	Start a single instance of nginx
<code>kubectl replace -f ./pod.json</code>	Force replace, delete and then re-create the resource. Will cause a service outage.
<code>kubectl edit svc/docker-registry</code>	Edit the service named docker-registry

### Resource types

<code>conf</code>	ConfigMaps allow you to decouple configuration artifacts from image content to keep containerized applications portable.
-------------------	--

### Resource types (cont)

<code>daemonsets</code>	A DaemonSet ensures that all (or some) Nodes run a copy of a Pod.
<code>deployments</code>	A Deployment controller provides declarative updates for Pods and ReplicaSets.
<code>ingresses</code>	An Ingress is a collection of rules that allow inbound connections to reach the cluster services.
<code>jobs</code>	A job creates one or more pods and ensures that a specified number of them successfully terminate.
<code>namespaces</code>	Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called namespaces.
<code>nodes</code>	Nodes are the physical cluster nodes.
<code>persistentvolumeclaims</code>	A PersistentVolumeClaim (PVC) is a request for storage by a user.
<code>persistentvolumes</code>	A PersistentVolume (PV) is a piece of storage in the cluster that has been provisioned by an administrator.
<code>Pods</code>	A Pod is the basic building block of Kubernetes—the smallest and simplest unit in the Kubernetes object model that you create or deploy.
<code>replicasets</code>	A ReplicaSet ensures that a specified number of pod replicas are running at any one time.
<code>cronjob</code>	A Cron Job manages time based Jobs.
<code>secrets</code>	Objects of type secret are intended to hold sensitive information.
<code>services</code>	A Kubernetes Service is an abstraction which defines a logical set of Pods and a policy by which to access them.