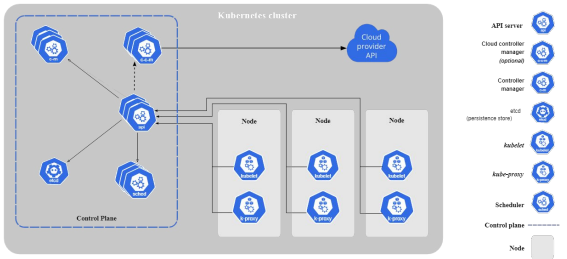
Kubernetes Cheat Sheet

Deleting resources	
Command	**What does it do?**
kubectl delete -f ./pod.json	Delete a pod using the type and name specified in pod.json
kubectl delete pod unwanted --now	Delete a pod with no grace period
kubectl delete pod,service baz foo	Delete pods and services with same names "baz" and "foo"
kubectl delete pods,services -l name=myLabel	Delete pods and services with label name=myLabel
kubectl -n my-ns delete pod,svc --all	Delete all pods and services in namespace my-ns
kubectl get pods -n mynamespace --no-headers=true | awk '/pattern1|pattern2/{print $1}' | xargs kubectl delete -n mynamespace pod	Delete all pods matching the awk pattern1 or pattern2

Interacting with resources	
Command	**What does it do?**
kubectl logs -l name=myLabel	dump pod logs, with label name=myLabel (stdout)
kubectl logs my-pod --previous	dump pod logs (stdout) for a previous instantiation of a container
kubectl logs my-pod -c my-container	dump pod container logs (stdout, multi-container case)
kubectl logs -l name=myLabel -c my-container	dump pod logs, with label name=myLabel (stdout)
kubectl logs my-pod -c my-container --previous	dump pod container logs (stdout, multi-container case) for a previous instantiation of a container
kubectl logs -f my-pod	stream pod logs (stdout)
kubectl logs -f my-pod -c my-container	stream pod container logs (stdout, multi-container case)
kubectl logs -f -l name=myLabel --all-containers	stream all pods logs with label name=myLabel (stdout)
kubectl run -i --tty busybox --image=busybox:1.28 -- sh	Run pod as interactive shell
kubectl run nginx --image=nginx -n mynamespace	Start a single instance of nginx pod in the namespace of mynamespace
kubectl run nginx --image=nginx	Run pod nginx and write its spec into a file called pod.yaml
kubectl top pod POD_NAME --containers	Show metrics for a given pod and its containers
kubectl attach my-pod -i	Attach to Running Container

Updating resources	
Command	**What does it do?**
kubectl set image deployment/frontend www=image:v2	Rolling update "www" containers of "frontend" deployment, updating the image
kubectl rollout history deployment/frontend	Check the history of deployments including the revision
kubectl rollout undo deployment/frontend --to-revision=2	Rollback to a specific revision
kubectl rollout status -w deployment/frontend	Watch rolling update status of "frontend" deployment until completion
kubectl rollout restart deployment/frontend	Rolling restart of the "frontend" deployment
cat pod.json | kubectl replace -f -	Replace a pod based on the JSON passed into stdin
kubectl replace --force -f ./pod.json	Force replace, delete and then re-create the resource. Will cause a service outage
kubectl expose rc nginx --port=80 --target-port=8000	Create a service for a replicated nginx, which serves on port 80 and connects to the containers on port 8000
kubectl get pod mypod -o yaml | sed 's/\(image: myimage\):.*$/\1:v4/' | kubectl replace -f -	Update a single-container pod's image version (tag) to v4

Formatting output	
Command	**What does it do?**
-o=custom-columns=<spec>	Print a table using a comma separated list of custom columns
-o=custom-columns-file=<filename>	Print a table using the custom columns template in the <filename> file
-o=json	Output a JSON formatted API object
-o=jsonpath=<template>	Print the fields defined in a jsonpath expression
-o=jsonpath-file=<filename>	Print the fields defined by the jsonpath expression in the <filename> file
cat pod.json | kubectl replace -f -	Replace a pod based on the JSON passed into stdin



Creating objects	
Command	**What does it do?**
kubectl apply -f ./my-manifest.yaml	create resource(s)
kubectl apply -f ./my1.yaml -f ./my2.yaml	create from multiple files
kubectl apply -f ./dir	create resource(s) in all manifest files in dir
kubectl apply -f https://git.io/vPieo	create resource(s) from url
kubectl create deployment nginx --image=nginx	start a single instance of nginx
kubectl explain pods	get the documentation for pod manifests

Viewing, finding resources	
Command	**What does it do?**
kubectl get services	List all services in the namespace
kubectl get pods --all-namespaces	List all pods in all namespaces
kubectl get pods -o wide	List all pods in the current namespace, with more details
kubectl get deployment my-dep	List a particular deployment
kubectl get pods	List all pods in the namespace
kubectl get pod my-pod -o yaml	Get a pod's YAML
kubectl describe nodes my-node kubectl describe pods my-pod	Describe commands with verbose output
kubectl get services --sort-by=.metadata.name	List Services Sorted by Name

Patching & Scaling resources	
Command	**What does it do?**
kubectl patch node k8s-node-1 -p '{"spec":{"unschedulable":true}}'	Partially update a node
kubectl patch pod valid-pod -p '{"spec":{"containers":[{"name":"kubernetes-serve-hostname","image":"new image"}]}}'	Update a container's image; spec.containers[*].name is required because it's a merge key
kubectl patch pod valid-pod --type='json' -p='[{"op": "replace", "path": "/spec/containers/0/image", "value":"new image"}]'	Update a container's image using a json patch with positional arrays
kubectl patch deployment valid-deployment --type json -p='[{"op": "remove", "path": "/spec/template/spec/containers/0/livenessProbe"}]'	Disable a deployment livenessProbe using a json patch with positional arrays
kubectl patch sa default --type='json' -p='[{"op": "add", "path": "/secrets/1", "value": {"name": "whatever" } }]'	Add a new element to a positional array
kubectl scale --replicas=3 rs/foo	Scale a replicaset named 'foo' to 3
kubectl scale --replicas=3 -f foo.yaml	Scale a resource specified in "foo.yaml" to 3
kubectl scale --current-replicas=2 --replicas=3 deployment/mysql	If the deployment named mysql's current size is 2, scale mysql to 3