**cheat sheet**

**kubectl syntax**

kubectl commands usually follow a fixed schema:

* a verb like get, describe, create, delete - tells kubectl what to do.
* a resource like pod, node - or any other valid resource - specifies for which resource type the action defined by the verb should be executed.
* a more detailed specification of the requested resource by ID/name or label (use -l <key=value>) lets you select one or more objects
* Parameters or modifiers like
  + -n=<your-namespace>: to route your query to another namespace than default
  + -o=wide: more detailed output. The flag also accepts yaml or json as format.

Example: kubectl get pod -n=my-namespace -o=wide

**short cuts**

kubectl wants you to reference resources by type. To do so, there are different formats available.

Most common is the following:  kubectl <verb> <resource type> <name>

Example: kubectl get deployment nginx

However, it is also possible to concatenate resource type and name like this: deployment/nginx.

For several of the resource types there are also shortened identifiers available. For example, the type service can be referenced by svc. A full list is available at <https://kubernetes.io/docs/reference/kubectl/>  in section "resource types".

To get a list of all resources of a cluster including their shortnames run: kubectl api-resources

Furthermore, it is possible to get all resources of different types belonging to a name or label:

kubectl get svc,pod,deploy -l name=nginx.

**other hints**

* kubectl explain <resource> will give you a description of the specified resource. To get more details append the field name to the resource like this: kubectl explain pod.spec
* --dry-run can be added to a kubectl command to run in simulation mode.
* kubectl diff -f <file> is useful to check & get the delta between a resource specified locally in a file and its currently deployed version in the cluster.

**most common kubectl commands**

Please note, the examples below are based on the “pod” resource, but also work with other resources like “service” or “deployment”.

| **command** | **expected result** |
| --- | --- |
| kubectl get all | get all objects of the most common resources (services, pods, ...) |
| kubectl get pods | get all pods in your current namespace |
| kubectl get pods  --show-labels=true | like the command above, but output shows all labels attached to the pods |
| kubectl get pods my-pod | get the pod my-pod |
| kubectl get pods -l status=awesome | get all pods with label "status=awesome” |
| kubectl describe pod  my-pod | gives a detailed description of the the pod my-pod. |
| kubectl create -f pod.yaml | create 1..n resources that are specified in the pod.yaml file |
| kubectl logs my-pod | prints the logs written by my-pod |
| kubectl exec -it  my-pod /bin/bash | starts a bash shell session in within the context of my-pod |
| kubectl label pod my-pod status=awesome | attaches a lable status=awesome to my-pod |
| kubectl delete pod my-pod | Deletes the pod my-pod. |

**Optionally, you can add a parameter “-n” to specify a dedicated namespace. If not specified, all requests will target the default namespace or the one specified in your kubeconfig.**