acend

kubectl cheat sheet

It's considered best practice to always set the namespace parameter.
This way you're sure you're modifying the correct resource.

Get information about resources

Display one or many resources in different formats:

```
kubectl -n NAMESPACE get TYPE [NAME] [-o yaml|wide]
```

Display labels of all resources of a specific type:

```
kubectl -n NAMESPACE get TYPE –show-labels
```

Show details of a specific resource or group of resources:

```
kubectl -n NAMESPACE describe TYPE [NAME]
```

Modify resources

Imperatively create the resource(s) defined in a file:

```
kubectl -n NAMESPACE create -f FILENAME
```

Helper commands for imperatively creating resources:

```
kubectl -n NAMESPACE create namespace|deployment|configmap|secret|ingress ...
```

Declaratively create or change the resource(s) defined in a file:

```
kubectl -n NAMESPACE apply -f FILENAME
```

Helper commands for defining environment variables or resource limits and requests:

```
kubectl -n NAMESPACE set env|resources ...
```

Redeploy an existing deployment or daemonset:

```
kubectl -n NAMESPACE rollout restart TYPE/NAME
```

Delete the resource(s) defined in a file:

```
kubectl -n NAMESPACE delete -f FILENAME
```

Delete a specific resource:

```
kubectl -n NAMESPACE delete TYPE NAME
```

Delete resources with a specific label:

```
kubectl -n NAMESPACE delete TYPE –selector=LABEL
```

Scale a deployment:

```
kubectl scale deployment/NAME –replicas=COUNT
```

Debugging and troubleshooting

Show a pod's logs:

```
kubectl -n NAMESPACE logs POD
```

Open a bash shell in a pod and attach to it:

```
kubectl -n NAMESPACE exec -it POD [-c CONTAINER] – /bin/bash
```

Forward one or more local ports to a pod:

```
kubectl -n NAMESPACE port-forward TYPE/NAME [LOCAL_PORT:]REMOTE_PORT
```

Copy directories and files from a container to a local directory:

```
kubectl cp NAMESPACE/POD:/tmp/foo /tmp/bar [-c CONTAINER]
```