

Kubernetes

Enjoy this cheat sheet at its fullest within Dash, the macOS documentation browser.

Creating Objects

Create resource(s) in a JSON or YAML file

```
kubectl create -f ./file.yml  
kubectl create -f ./file1.yml -f ./file2.yaml
```

Create resources in all .json, .yaml, and .yml files in dir

```
kubectl create -f ./dir
```

Create from a URL

```
kubectl create -f http://www.fpaste.org/279276/48569091/raw/
```

Create multiple YAML objects from stdin

```
cat <<EOF | kubectl create -f -  
apiVersion: v1  
kind: Pod  
metadata:  
  name: busybox-sleep  
spec:  
  containers:  
  - name: busybox  
    image: busybox  
    args:  
    - sleep  
    - "1000000"  
---  
apiVersion: v1  
kind: Pod  
metadata:  
  name: busybox-sleep-less  
spec:  
  containers:  
  - name: busybox  
    image: busybox  
    args:  
    - sleep  
    - "1000"  
EOF
```

Create a secret with several keys

```
cat <<EOF | kubectl create -f -  
apiVersion: v1  
kind: Secret
```

```
metadata:  
  name: mysecret  
type: Opaque  
data:  
  password: $(echo "s33msi4" | base64)  
  username: $(echo "jane" | base64)  
EOF
```

Viewing and Finding Resources

List all namespaces in the cluster

```
kubectl get namespaces
```

List all services in the namespace

```
kubectl get services
```

List all pods in all namespaces

```
kubectl get pods --all-namespaces
```

List all pods in the namespace, with more details

```
kubectl get pods -o wide
```

List a particular replication controller

```
kubectl get rc <rc-name>
```

List a particular RC

```
kubectl get replicationcontroller <rc-name>
```

List a particular node with verbose output

```
kubectl describe nodes <node-name>
```

List a particular pod with verbose output

```
kubectl describe pods <pod-name>  
kubectl describe pods/<pod-name> # Equivalent to previous
```

Lists pods created by using common prefix

```
kubectl describe pods <rc-name>
```

List services sorted by name

```
kubectl get services --sort-by=.metadata.name
```

List pods sorted by restart count

```
kubectl get pods --sort-by=.status.containerStatuses[0].restartCount
```

Get the version label of all pods with label app=cassandra

```
kubectl get pods --selector=app=cassandra rc -o 'jsonpath={.items[*].metadata.labels.version}'
```

Get ExternalIPs of all nodes

```
kubectl get nodes -o jsonpath='{.items[*].status.addresses[?(@.type=="ExternalIP")].address}'
```

List names of pods that belong to Particular RC

```
# "jq" command useful for transformations that are too complex for jsonpath
sel=$(./kubectl get rc <rc-name> --output=json | jq -j '.spec.selector | to_entries | .[] | "\(.key)=\(.value),"'
sel=${sel%?} # Remove trailing comma
pods=$(kubectl get pods --selector=$sel --output=jsonpath={.items..metadata.name})`
```

Check which nodes are ready

```
kubectl get nodes -o jsonpath='{range .items[*]}{@.metadata.name}:{range @.status.conditions[*]}{@.type}={@.status};{end}{end}}' | tr ';' '\n' | grep "Ready=True"
```

Modifying and Deleting Resources

Add a label

```
kubectl label pods <pod-name> new-label=awesome
```

Add an annotation

```
kubectl annotate pods <pod-name> icon-url=http://goo.gl/XXBTWq
```

Interacting with Running Pods

Dump pod logs (stdout)

```
kubectl logs <pod-name>
```

Stream pod logs (stdout) until canceled (ctrl-c) or timeout

```
kubectl logs -f <pod-name>
```

Run pod as interactive shell

```
kubectl run -i --tty busybox --image=busybox -- sh
```

Attach to running container

```
kubectl attach <podname> -i
```

Forward port of pod to your local machine

```
kubectl port-forward <podname> <local-and-remote-port>
```

Forward port to service

```
kubectl port-forward <servicename> <port>
```

Run command in existing pod (1 container case)

```
kubectl exec <pod-name> -- ls /
```

Run command in existing pod (multi-container case)

```
kubectl exec <pod-name> -c <container-name> -- ls /
```

Notes

- Adapted from the official Kubernetes documentation cheatsheet: <http://kubernetes.io/docs/user-guide/kubectl-cheatsheet/>.

You can modify and improve this cheat sheet [here](#)