kubernetes

CONTINO

Kubernetes Command Cheat Sheet

Context Configuration

```
kubectl config view                                     # Display kubeconfig
kubectl config current-context                          # Display current context
kubectl config use-context "contextname"                # Switch context
Kubectl config set-context "context_name" -n "namespace"  # Set new context
```

Creating and Scaling Objects

```
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 and svc
                                                        manifests
kubectl run nginx --image=nginx --replicas=5          # Creates a deployment with 5 replicas
kubectl run --generator=run-pod/v1 --image=nginx \    # Prints the corresponding apis without
--dry-run POD_NAME                                      creating the Objects
kubectl expose pod pod-name \                         # Creates a service for the pod and
--port=444 --name=frontend                              exposes port 444 on the pod and gives
                                                        it a name frontend

kubectl autoscale deployment DEPLOYMENT \
    --min=# --max=# --cpu-percent=#                   # Create an autoscaler for a deployment
```

Viewing Resources

```
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 namespace, with more details
kubectl get deployment my-dep               # List a particular deployment
kubectl get pods --include-uninitialized    # List all pods in the namespace, including uninitialized
                                              ones
```

Describe Commands with Verbose Output

```
kubectl describe nodes my-node
kubectl describe pods my-pod
```

Cluster Events

```
kubectl get events
```

Updating Resources

```
kubectl set image deployment/nginx nginx=nginx:1.9.2         # Rolling update of the deployment
                                                               nginx, updating the image from
                                                               nginx to nginx:1.9.2
kubectl rollout undo deployment/frontend                     # Rollback to the previous deployment
kubectl rollout status -w deployment/frontend                # Watch rolling update status of
                                                               "frontend" deployment until
                                                               completion
kubectl rollout history deployment/nginx                     # See previous rollouts of deployment
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 label pods my-pod new-label=awesome                  # Add a label
kubectl annotate pods my-pod icon-url=http://goo.gl/abc      # Add an annotation
kubectl autoscale deployment foo --min=2 --max=10            # Auto scale a deployment
                                                               "foo"
```

View Logs, Exec Into Container and Troubleshooting

```
kubectl logs  pod-name                                              # Prints logs from stdout
                                                                      produced by pod-name
kubectl logs  pod-name -c container-name                            # When you have multiple
                                                                      containers in the pod
kubectl get pods pod-name -o jsonpath={.spec.containers[*].name}    # Lists all the containers in the
                                                                      same pod
kubectl exec -it "podname"-- /bin/bash                              # Executes into container and
                                                                      opens the shell, mainly used
                                                                      for troubleshooting the
                                                                      container
kubectl exec -it "podname" -c "containername" -- /bin/bash          # Allows you to shell into a
                                                                      specific container within a pod
```

Deleting Any Unwanted Kubernetes Objects or Resources

```
kubectl delete pods pod-name
kubectl delete deployment deployment-name
kubectl delete
deployment,service,ingress,networkpolicy,configmap,secret --selector run=nginx
```

Resources

* certificatesigningrequests (aka 'csr')
* clusterrolebindings
* clusterroles
* componentstatuses (aka 'cs')
* configmaps (aka 'cm')
* controllerrevisions
* cronjobs
* customresourcedefinition (aka 'crd')
* daemonsets (aka 'ds')
* deployments (aka 'deploy')
* endpoints (aka 'ep')
* events (aka 'ev')
* horizontalpodautoscalers (aka 'hpa')
* ingresses (aka 'ing')
* jobs
* limitranges (aka 'limits')
* namespaces (aka 'ns')
* networkpolicies (aka 'netpol')
* nodes (aka 'no')
* persistentvolumeclaims (aka 'pvc')
* persistentvolumes (aka 'pv')
* poddisruptionbudgets (aka 'pdb')
* podpreset
* pods (aka 'po')
* podsecuritypolicies (aka 'psp')
* podtemplates
* replicasets (aka 'rs')
* replicationcontrollers (aka 'rc')
* resourcequotas (aka 'quota')
* rolebindings
* roles
* secrets
* serviceaccounts (aka 'sa')
* services (aka 'svc')
* statefulsets (aka 'sts')
* storageclasses (aka 'sc')