Virtual Pair Programmers

Kubernetes - Quick Command Reference 1

Here's a run down of all (ok, most of) of the commands used in the Kubernetes Microservices course

Minikube

minikube start

starts minikube. If this hangs (wait 15 minutes), then see the video in section 3 that addresses common problems

minikube stop

(Section 3): stops the minikube virtual machine. This may be necessary to do if you have an error when starting

minikube delete

(Section 3): do this to completely wipe away the minikube image. Useful if minikube is refusing to start and all else fails. Also you can delete all files in <home>/.minikube and <home>/.kube

minikube env

(Section 4): find out the required environment variables to connect to the docker daemon running in minikube.

minikube ip

(Section 4 or 5): find out the ip address of minikube. Needed for browser access.

Kubectl

kubectl get all

(Section 5): list all objects that you've created. Pods at first, later, ReplicaSets, Deployments and Services

kubectl apply –f <yaml file>

(Section 5): either creates or updates resources depending on the contents of the yaml file

kubectl apply –f .

(Section 7): apply all yaml files found in the current directory

Kubernetes - Quick Command Reference 2

kubectl describe pod <name of pod>
(Section 5): gives full information about the specified pod

kubectl exec –it <pod name> <command>
(Section 5): execute the specified command in the pod's container.
Doesn't work well in Cygwin.

kubectl get (pod | po | service | svc | rs | replicaset | deployment | deploy)
(Section 6): get all pods or services. Later in the course, replicasets and deployments.

kubectl get po --show-labels
(Section 6): get all pods and their labels

kubectl get po --show-labels -l {name}={value}
(Section 6): get all pods matching the specified name:value pair

kubectl delete po <pod name>
(Section 8): delete the named pod. Can also delete svc, rs, deploy

kubectl delete po --all
(Section 8): delete all pods (also svc, rs, deploy)

Deployment Management

kubectl rollout status deploy <name of deployment>
(Section 9): get the status of the named deployment

kubectl rollout history deploy <name of deployment>
(Section 9): get the previous versions of the deployment

kubectl rollout undo deploy <name of deployment>
(Section 9): go back one version in the deployment. Also optionally --to-revision=<revision number>
We recommend this is used only in stressful emergency situations! Your YAML will now be out of date with the live deployment!