Kubernetes Cheat Sheet

Pods

```
# Get all pods in the current namespace
kubectl get pods

# Get pods in all namespaces
kubectl get pods --all-namespaces

# Get pods with more details
kubectl get pods -o wide

# Get the yaml for a pod
kubectl get pod <pod> -o yaml

# Inspect a pod
kubectl describe pods <pod>

# Get pods sorted by a metric
kubectl get pods \
  --sort-by='.status.containerStatuses[0].restartCount'

# Get pods with their labels
kubectl get pods --show-labels

# Get pods that match a label
kubectl get pods -l <label>=<value>

# Forward traffic from a localhost port to a pod port
kubectl port-forward <pod> <localhost-port>:<pod-port>

# Run a command on a pod
kubectl exec <pod> -- <command>

# Run a command on a container in a pod
kubectl exec <pod> -c <container> -- <command>
```

Secrets

```
# Get all secrets in the current namespace
kubectl get secrets

# Get secrets in all namespaces
kubectl get secrets --all-namespaces

# Get secrets with more details
kubectl get secrets -o wide

# Get the contents of a secret
kubectl get secrets <secret> -o yaml
```

Services

```
# Get all services in the current namespace
kubectl get services

# Get services in all namespaces
kubectl get service --all-namespaces

# Get services with more details
kubectl get service -o wide

# Get the yaml for a services
kubectl get service <service> -o yaml

# Inspect a service
kubectl describe service <service>

# Get service's labels
kubectl get service --show-labels

# Get services that match a label
kubectl get service -l <label>=<value>
```

Updating Resources

```
# Roll a new version of a deployment
kubectl set image deployment/<deployment> <container-
name>=image:<version>

# Check the deployment history
kubectl rollout history deployment/<deployment>

# Rollback a deployment
kubectl rollout undo deployment/<deployment>

# Rollback to a specific version
kubectl rollout undo deployment/<deployment> --to-revision=2

# Watch a rolling update
kubectl rollout status -w deployment/<deployment>

# Restart the rolling deploy
kubectl rollout restart deployment/<deployment>

# Edit a resource’s yaml
kubectl edit deployment/<deployment>

# Scale a deployment to 3 pods
kubectl scale --replicas=3 deployment/<deployment>

# Delete a pod
kubectl delete pod <pod>
```

Context

```
# Show contexts
kubectl config get-contexts

# Show current context
kubectl config current-context

# Switch context to another cluster
kubectl config use-context <my-cluster-name>

# Change Namespace
kubectl config set-context --current --namespace=<namespace>
```

Logs

```
# Show logs (stdout) of a pod
kubectl logs <pod>

# Show logs (stdout) of pods that match a label
kubectl logs -l <label>=<value>

# Show logs of a previous instantiation of a container
kubectl logs <pod> --previous

# Show logs for a specific container in a pod (i.e. init
container)
kubectl logs <pod> -c <container>

# Following logs from a pod
kubectl logs -f <pod>

# Follow all logs from a pod that match a label
kubectl logs -f -l <label>=<value> --all-containers

# Show logs with verbosity level of logs from 0 - 9
kubectl logs <pod> --v=<0:9>
```

Deployments

```
# Get all deployments in the current namespace
kubectl get deployment

# Get deployments in all namespaces
kubectl get deployment --all-namespaces

# Get deployments with more details
kubectl get deployment -o wide

# Get the yaml for a deployment
kubectl get deployment <deployment> -o yaml

# Inspect a deployment
kubectl describe deployment <deployment>

# Get deployment's labels
kubectl get deployment --show-labels

# Get deployments that match a label
kubectl get deployment -l <label>=<value>
```

Ingress

```
# Get all ingress in the current namespace
kubectl get ingress

# Get ingress in all namespaces
kubectl get ingress --all-namespaces

# Get ingress with more details
kubectl get ingress -o wide

# Get the yaml for a ingress
kubectl get ingress <ingress> -o yaml

# Inspect a ingress
kubectl describe ingress <ingress>

# Get ingress labels
kubectl get ingress --show-labels

# Get ingress that match a label
kubectl get ingress -l <label>=<value>
```

Creating Resources

```
# Create a kubernetes resource from a file
kubectl apply -f ./<manifest>.yaml

# Create kubernetes resources from multiple files
kubectl apply -f ./<manifest>.yaml -f ./<manifest>.yaml

# Create resources from all manifest files in a directory
kubectl apply -f ./<directory>

# Create resource from a url
kubectl apply -f <url_to_manifest>

# Start a single instance of an image
kubectl create deployment <deployment_name> --
image=<image>
```

Nodes

```
# Mark node as unschedulable
kubectl cordon <node>

# Drain a node for maintenance
kubectl drain <node>

# Mark node as schedulable
kubectl uncordon <node>

# Show ‘top’ metrics for a node
kubectl top node <node>

# Display addresses of the master and services
kubectl cluster-info

# Dump current cluster state to stdout
kubectl cluster-info dump

# Show a list of eligible kube resource (i.e. pods, service, pv,
etc)
kubectl api-resources

# Show a list of eligible kube resources in your namespace
kubectl api-resources --namespaced=true
```