Kubernetes Cheat Sheet

Installation

Install the `kubectl` command line tool to interact with the Kubernetes API:
https://kubernetes.io/docs/tasks/tools/#kubectl

Enable autocompletion in bash:

```
composer completion bash | sudo tee /etc/bash_completion.d/kubectl
```

Global flags

Flag	Description
`--namespace <namespace>`	The name of the namespace to use
`--context <context>`	The name of the context to use
`--help`	Show information about a given command

Context and configuration

Command	Description
`kubectl config get-contexts`	List all contexts
`kubectl config current-context`	Display the current context
`kubectl config use-context <context>`	Switch to another context
`kubectl config delete-context <context>`	Delete the specified context from the kubeconfig

Display resources

Command	Description
`kubectl get <resource>`	List all resources of this type in the current namespace
`kubectl get <resource> -o wide`	List all resources with more details
`kubectl get <resource> -A`	List all resources of this type in all namespaces
`kubectl get <resource> <name>`	List a particular resource
`kubectl get <resource> <name> -o yaml`	Print a particular resource in YAML format
`kubectl get <resource> <name> -l <key1>=<value1>`	List resources where label <key1> contains <value1>
`kubectl describe <resource>`	Show detailed information about a resource

Apply configuration manifests

Command	Description
`kubectl apply -f <file>`	Apply a manifest from a file
`kubectl apply -f <dir>`	Apply all manifests in a directory
`kubectl apply -k <dir>`	Apply resources from a kustomize directory

Create resources manually

Command	Description
`kubectl run <name> --image=<image>`	Start a pod
`kubectl create deployment <name> --image=<image>`	Create a deployment
`kubectl expose pod <pod> --port=<port>`	Create a service for an existing pod
`kubectl expose deployment <name> --port=<port>`	Create a service for an existing deployment
`kubectl create ingress <name> --rule=<host/path=svc:port>`	Create an ingress that routes traffic to a service
`kubectl create job <name> --image=<image>`	Create a job
`kubectl create job <name> --from=cronjob/<name>`	Create a job from a cronjob
`kubectl create cronjob <name> --image=<image> --schedule=<schedule>`	Create a cronjob, using a schedule in Cron format
`kubectl create secret generic <name> --from-literal=<key>=<value>`	Create a secret containing <key> and <value>
`kubectl create secret docker-registry <name> --docker-server=<server> --docker-username=<username> --docker-password=<password>`	Create a secret for a Docker registry

Generate YAML configuration manifests

Command	Description
`kubectl create deployment <name> --image=<image> --dry-run=client -o yaml`	Generate a deployment manifest
`kubectl expose deployment <name> --port=<port> --dry-run=client -o yaml`	Generate a service manifest for a deployment

Edit resources

Command	Description
`kubectl edit <resource> <name>`	Edit a resource in a text editor
`kubectl set image <resource> <name> <container>=<image>`	Update the image of a container in a pod

Set labels and annotations

Command	Description
`kubectl label <resource> <name> <key>=<value>`	Add a label to a resource
`kubectl annotate <resource> <name> <key>=<value>`	Add an annotation to a resource

Delete resources

Command	Description
`kubectl delete <resource> <name>`	Delete a particular resource
`kubectl delete <resource> --all`	Delete all resources of a particular type in the current namespace
`kubectl delete -f <file>`	Delete a resource from a file

Manage deployments

Command	Description
`kubectl rollout status deployment <name>`	Show the status of a deployment rollout
`kubectl rollout history deployment <name>`	View the rollout history of a deployment
`kubectl rollout undo deployment <name>`	Undo a previous rollout deployment
`kubectl rollout restart deployment <name>`	Restart a deployment
`kubectl scale deployment <name> --replicas=<n>`	Scale a deployment to <n> replicas
`kubectl autoscale deployment <name> --min=<min> --max=<max>`	Autoscale a deployment between <n> and <n> replicas

Execute commands

Command	Description
`kubectl exec <pod> -- <command>`	Execute a command in a running pod
`kubectl exec -it <pod> -- sh`	Open a shell in a running pod

View logs

Command	Description
`kubectl logs <pod>`	Print the logs for a pod
`kubectl logs -f <pod>`	Print the logs for a pod and keep streaming

Resource usage

Command	Description
`kubectl top node`	Show resource (CPU/memory) usage of nodes
`kubectl top pod`	Show resource (CPU/memory) usage of pods

Other commands

Command	Description
`kubectl version`	Show the version of the client and server
`kubectl api-resources`	Print the supported API resources on the server

Helm

Helm is the package manager for Kubernetes. See https://helm.sh/docs/intro/install/ for installation instructions.

Global Helm flags

Flag	Description
`--kube-context <name>`	Name of the Kubernetes context to use
`--namespace <name>`	Namespace to use for this operation

Helm repository management

Command	Description
`helm repo add <name> <url>`	Add a repository
`helm repo list`	List all added repositories
`helm repo update`	Update the local cache of available charts
`helm repo remove <name>`	Remove a repository
`helm search repo`	List all charts in the repositories
`helm search repo <keyword>`	Search for a chart in the repositories

Installing Helm charts

Command	Description
`helm install <name> <chart>`	Install a chart with a name
`helm install <chart> --generate-name`	Install a chart, auto-generating a name
`helm install <name> <chart> --namespace <namespace>`	Install a chart in a specific namespace
`helm install <name> <chart> --set <key>=<value>`	Install a chart with specific values
`helm install <name> <chart> --values <file>`	Install a chart using a values file
`helm install <name> <chart> --dry-run --debug`	Run a test installation to validate the chart
`helm install <name> <chart> --verify`	Verify the package before installing
`helm install <name> <chart> --dependency-update`	Update dependencies before installing
`helm uninstall <name>`	Uninstall a release
`helm uninstall <name> --keep-history`	Uninstall a release, keeping the history

Listing Helm releases

Command	Description
`helm list`	List all releases in the current namespace
`helm list --all-namespaces`	List all releases in all namespaces
`helm list -l <label>=<value>`	List releases with a specific label
`helm list --date`	List releases sorted by date
`helm list --(pending|failed|uninstalled)`	List releases that are in a pending/failed/uninstalled state
`helm status <name>`	Show the status of a release

Managing Helm releases

Command	Description
`helm upgrade <name> <chart>`	Upgrade a release
`helm upgrade <name> <chart> --atomic`	Upgrade a release atomically
`helm upgrade <name> <chart> --dependency-update`	Upgrade a release and update dependencies
`helm upgrade <name> <chart> --version <version>`	Upgrade a release to a specific version
`helm upgrade <name> <chart> --set <key>=<value>`	Upgrade a release with specific values
`helm rollback <release> <revision>`	Rollback a release to a previous revision

Developing Helm charts

Command	Description
`helm create <name>`	Create a new chart
`helm package <chart-path>`	Package a chart directory into a chart file
`helm lint <chart>`	Lint a chart
`helm show all <chart>`	Inspect a chart and list all resources
`helm show values <chart>`	Inspect a chart and show default values
`helm template <name> <chart>`	Render templates locally
`helm template <name> <chart> --set <key>=<value>`	Render templates locally and override values