

kubectl 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
<code>--namespace <namespace></code>	The name of the namespace to use
<code>--context <context></code>	The name of the context to use
<code>--help</code>	Show information about a given command

Context and configuration

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

Display resources

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

Apply configuration manifests

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

Create resources manually

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

Generate YAML configuration manifests

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

Edit resources

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

Set labels and annotations

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

Delete resources

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

Manage deployments

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

Execute commands

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

View logs

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

Resource usage

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

Other commands

Command	Description
<code>kubectl version</code>	Show the version of the client and server
<code>kubectl api-resources</code>	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
<code>--kube-context <name></code>	Name of the Kubernetes context to use
<code>--namespace <name></code>	Namespace to use for this operation

Helm repository management

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

Installing Helm charts

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

Listing Helm releases

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

Managing Helm releases

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

Developing Helm charts

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