

# ToiletPaper #102

## Helm - The package manager for Kubernetes

Author: Dr. Torsten Ackemann, Senior Systems Architect, Competence Center Platforms & Operations

### ✘ Problem

How do I install standard applications in Kubernetes? How do I configure these? Who manages dependencies and upgrades? And: How do I bring my own application into a Kubernetes? It was so easy with docker-compose!

### ✓ Solution

Helm is a package manager that allows to install complex applications such as Wordpress, Hadoop or Gitlab onto Kubernetes clusters – including dependencies such as a database.

In principle, Helm is to Kubernetes what yum or aptitude is to Linux. But a package can be installed multiple times.

The packages are called Helm charts and consist of the Kubernetes resources as templates in YAML format, a configuration file values.yaml with the default settings of the chart, a list of dependencies to other Helm charts and some more metadata such as name and version.

Further configuration can be handed over to Helm in the form of additional YAMLS which then overwrite the defaults.

Concrete resource descriptions are generated from the templates and the default values as well as from the transferred configuration and are sent to Kubernetes. Kubernetes then creates resources such as docker containers, storage volumes and so on.

With your own Helm chart for self-written applications, you can put a description of the required runtime conditions into code – how many instances should run, what are the memory requirements, what other containers should run in parallel as sidecars, which databases are needed and how should they be configured, which endpoints should be reachable from the outside, should the application run permanently or only daily at 4 am, how to determine whether the application is still running and when is it ready to receive traffic etc. – and that's of course all configurable for different types of environments.

Deployments in the dev environment on the laptop and deployments in the prod environment in the cloud may differ only in the configuration YAML.

### ➔ Example

The following commands start Minikube (Kubernetes in a VM) on your laptop, install a complete Wordpress with extra memory plus database, storage volume and load balancer and display the Wordpress page in your browser:

```
$ minikube start
Starting local Kubernetes v1.10.0 cluster...
$ helm init
Adding stable repo with URL: https://kubernetes-charts.storage.googleapis.com
$ cat my-values.yaml
resources:
  requests:
    memory: 1024Mi
$ helm install -f my-values.yaml stable/wordpress
==> v1/Service
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
inclined-woodpecker-mariadb        ClusterIP           10.110.172.10   <none>           3306/TCP         0s
inclined-woodpecker-wordpress      LoadBalancer       10.108.66.217   <pending>        80:30095/TCP,443:31876/TCP    0s
$ minikube service inclined-woodpecker-wordpress
Opening kubernetes service default/inclined-woodpecker-wordpress in default browser...
```

*inclined-woodpecker* is an automatically generated name, so that you could install a second Wordpress without a name clash.

### + Further Aspects

- [www.helm.sh](http://www.helm.sh) – the official documentation
- [github.com/helm/charts](https://github.com/helm/charts) – the official Helm Charts e.g. for Wordpress
- [coreos.com/operators](https://coreos.com/operators) – Kubernetes Operators: For those who find Helm Charts too static