

Verteilte Systeme & Cloud- Technologien

HELM Charts

Kubernetes Package Manager



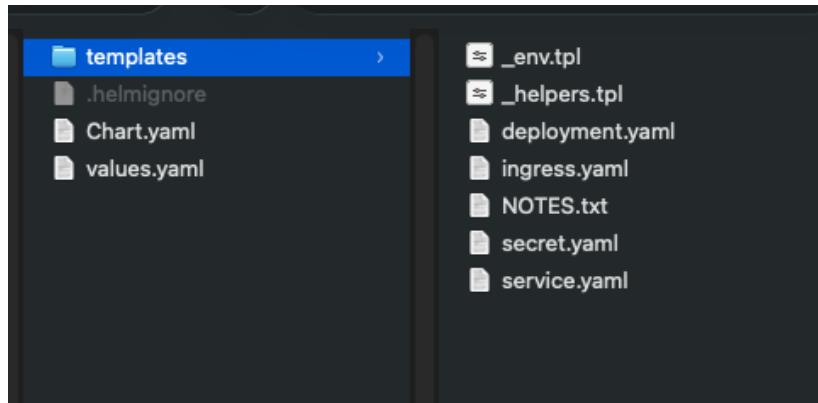
Was ist HELM?

- **Package Manager** für Kubernetes - wie apt/yum für Linux
- Charts = vorkonfigurierte Kubernetes Resource Templates
- Versionierung und Rollback von Deployments
- Dependency Management zwischen Services
- Wiederverwendbare Konfigurationen



<https://helm.sh/>

Chart Struktur



- **Chart.yaml** - Metadaten (Name, Version, Dependencies)
- **values.yaml** - Default Konfigurationswerte
- **templates/** - Kubernetes Resource Templates
 - **deployment.yaml,** **service.yaml,** **configmap.yaml**
 - **_helpers.tpl** - Wiederverwendbare Template-Snippets
- **charts/** - Abhängige Charts

<https://helm.sh/>

Wichtige Befehle

- `helm create <name>` - Neues Chart erstellen
- `helm install <release> <chart>` - Chart deployen
- `helm upgrade <release> <chart>` - Update durchführen
- `helm rollback <release>` - Zur vorherigen Version
- `helm list` - Deployments anzeigen
- `helm uninstall <release>` - Deployment entfernen

<https://helm.sh/>



Templating

- Go Template Engine
 - `{{ .Values.replicaCount }}` - Wert aus values.yaml
 - `{{ .Release.Name }}` - Release-spezifische Variablen
 - Conditionals: `{{ if .Values.ingress.enabled }}`
 - Loops: `{{ range .Values.containers }}`
 - Helper Functions: `{{ include "chart.labels" . }}`

<https://helm.sh/>



Best Practices

- Semantic Versioning für Charts verwenden
 - **values.yaml** gut dokumentieren
 - **helm lint** vor Release ausführen
 - **--dry-run** für Validierung nutzen
 - Sensitive Data über Secrets, nicht in **values.yaml**
 - Repository für Team-Charts einrichten

<https://semver.org/>

<https://helm.sh/>



Quick Start

- Installation: `brew install helm` (Mac) / `choco install kubernetes-helm` (Win) -> <https://helm.sh/docs/intro/install/>

- Repo hinzufügen und chart suchen / holen

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

```
helm search repo <keyword>
```

```
helm pull bitnami/postgresql --untar
```

- Anpassen: `values.yaml` editieren oder `--set` parameter

```
helm install my-db bitnami/postgresql -f custom-values.yaml
```



Vorteile für Teams

- Environment-spezifische Deployments (dev/stage/prod)
- CI/CD Integration mit GitOps
- Komplexe Microservice-Stacks mit **einem Befehl**
- **Rollback** bei Problemen in Sekunden
- Team-übergreifende Chart-Bibliotheken
- **Reduzierte Fehlerquote** durch Standardisierung

