

官方文档: [https://helm.sh/zh/docs/chart\\_template\\_guide/getting\\_started/](https://helm.sh/zh/docs/chart_template_guide/getting_started/)

## helm命令

```
1 #列出已添加的仓库
2 helm repo list
3 #更新repo
4 helm repo update
5 #从仓库搜索mysql
6 helm search repo mysql
7 #安装
8 helm install [NAME] [CHART] -f values.yaml
9 #查看已安装服务
10 helm list -n dev
11 #查看已安装value配置
12 helm get values -n dev [NAME]
13 #查看完整yaml
14 helm get manifest NAME
15 #更新
16 helm upgrade --dry-run -n dev -f ./mysql-dev/values.yaml [NAME] ./mysql-dev/
17 #查看版本号
18 helm history [NAME]
19 #回滚
20 helm rollback [NAME] [REVISION]
21 #卸载
22 helm uninstall [NAME]
23
24 #调试
25 helm install --debug --dry-run NAME PATH
```

## Chart 结构:

```
1 Chart.yaml          # 包含了chart信息的YAML文件
2 values.yaml         # chart 默认的配置值
3 charts/             # 包含chart依赖的其他chart
4 crds/               # 自定义资源的定义, CRD被视为一种特殊的对象。它们被安装在ch
5 templates/          # 模板目录, 当和values 结合时, 可生成有效的Kubernetes mar
6 templates/_helpers.tpl: #放置可以通过chart复用的模板辅助对象
```

## Chart.yaml 文件

- 1 `apiVersion` 字段
- 2 对于至少需要Helm 3的chart, `apiVersion` 字段应该是 `v2`。Chart支持之前`apiVersion` 设置
- 3 `v1` 到 `v2`的改变:
- 4 `dependencies`字段定义了chart的依赖, 针对于`v1` 版本的chart被放置在分隔开的`requiremer`
- 5 `type`字段, 用于识别应用和库类型的chart (查看 `Chart` 类型)。

- 1 #创建一个chart
- 2 `helm create mychart`
- 3 #校验
- 4 `helm lint mychart`
- 5 #打包 `mychart-0.1.-.tgz`
- 6 `helm package mychart`

- 1 #常用
- 2 #双引号
- 3 `{{ .Values.favorite.drink | quote }}`
- 4 #默认值
- 5 `{{ .Values.favorite.drink | default "tea" | quote }}`
- 6 `{{ .Values.favorite.drink | default (printf "%s-tea" (include "fullname" .))`

- 1 #自定义模板
- 2 `{{- define "mychart.app" -}}`
- 3 `app_name: {{ .Chart.Name }}`
- 4 `app_version: "{{ .Chart.Version }}"`
- 5 `{{- end -}}`
- 6 #引用 (无缩进)
- 7 `{{ include "mychart.app" . | indent 2 }}`
- 8 #引用 (原格式缩进)
- 9 `{{ template "mychart.app" . }}`

- 1 #流程控制
- 2 `{{ if PIPELINE }}`
- 3 `# Do something`

```

4 {{ else if OTHER PIPELINE }}
5   # Do something else
6 {{ else }}
7   # Default case
8 {{ end }}
9 #样例
10 {{- if .Values.replicaCount }}
11 replicas: {{ .Values.replicaCount }}
12 {{- end }}
13
14 #range 当值为字典
15 {{- range $index, $topping := .Values.kv }}
16   {{ $index }}: {{ $topping }}
17 {{- end }}
18 # 当值为列表
19 {{- range .Values.pizzaToppings }}
20 - {{ . | title | quote }}
21 {{- end }}
22
23 #以下值为 false:
24 #   布尔false
25 #   数字0
26 #   空字符串
27 #   nil (空或null)
28 #   空集合(map, slice, tuple, dict, array)

```

**{{- 3 }}** 表示“删除左边空格并打印3”，而**{{-3 }}**表示“打印-3”。要确保-和其他命令之间有一个空格

```

1 ## .是对当前作用域的引用
2 ## with允许你为特定对象设定当前作用域(.)
3 #样例
4 {{- with .Values.nodeSelector }}
5     nodeSelector:
6     {{ toYaml . | indent 8 }} ##将.Values.nodeSelector的值用解析为yaml
7     {{- end }}
8

```

## 内置变量

## Release

- .Release.Name: release名称
- .Release.Namespace 以外部传递NS为准，不会读取模板内namespace
- .Release.IsUpgrade: 如果当前操作是升级或回滚的话，需要将该值设置为true
- .Release.IsInstall: 如果当前操作是安装的话，需要将该值设置为true
- .Release.Revision 此次修订的版本号。安装时是1，每次升级或回滚都会自增
- .Release.Service: 该service用来渲染当前模板。Helm里一般是Helm

## Values values.yaml文件传递

### Chart: Chart.yaml文件内容（一般为Helm项目信息）

Files: 在chart中提供访问所有的非特殊文件。当你不能使用它访问模板时，你可以访问其他文件。请查看这个 [文件访问部分](#) 了解更多信息

Files.Get 通过文件名获取文件的方法。 (.Files.Getconfig.ini)

Files.GetBytes 用字节数组代替字符串获取文件内容的方法。对图片之类的文件很有用

Files.Glob 用给定的shell glob模式匹配文件名返回文件列表的方法

Files.Lines 逐行读取文件内容的方法。迭代文件中每一行时很有用

Files.AsSecrets 使用Base 64编码字符串返回文件体的方法

Files.AsConfig 使用YAML格式返回文件体的方法

## Values使用

清空values文件中的字段并重新写入

```
--set livenessProbe.httpGet=null --set livenessProbe.exec.command=[cat,/docroot/CHANGELOG.txt]
```

## Helm 逻辑和流控制函数

and 返回两个参数的and布尔值

```
1 and .Arg1 .Arg2
```

or 会返回第一个非空参数或最后一个参数

```
1 or .Arg1 .Arg2
```

not 布尔取反 （非空值、非0为true）

```
1 not .Arg
```

eq ne 等于、不等于

```
1 eq .Arg1 .Arg2
```

lt le 小于、小于等于、 gt ge 大于、大于等于

```
1 lt .Arg1 .Arg2
```

empty 判断是否为空

```
1 empty .Foo
```

coalesce 返回第一个非空值

```
1 coalesce 0 1 2#返回1
2 {{ coalesce .name .parent.name "Matt" }}
```

ternary 获取两个值和一个test值。如果test值是true，则返回第一个值。如果test值是空，则返回第二个值 {{ true | ternary "foo" "bar" }}

default 设定默认值

```
1 default "foo" .Bar
```

**"空"定义取决于以下类型：**

整形: 0

字符串: ""

列表: []

字典: {}

布尔: false

以及所有的nil (或 null)

字符处理函数：

print 打印

```
1 {{ print "Matt has " .Values.test.d " dogs" | quote }}
```

println 打印，末尾换行

printf 传参打印

```
1 printf "%s has %d dogs." .Name .NumberDogs
```

trim 去除2边空格

trimAll 移除字符

```
1 {{ trimAll "$" "$5.00" }} ==> 5.00
```

trimPrefix 字符串中移除前缀

trimSuffix 从字符串中移除后缀

lower/upper 转换大小写

title/untitle 首字母大写、首字母小写

```
1 title "hello world" ---> "Hello World"
```

repeat 重复多次

substr 获取字符串的子串

```
1 substr 0 5 "hello world" -> hello
```

nospace 去掉字符串中的所有空格

trunc 截断字符串

```
1 trunc 5 "XXXXX"
2 trunc -5 "XXXXX" #倒数5个
```

contains 测试字符串是否包含在另一个字符串中

```
1 contains "cat" "catch"
```

hasPrefix 和 hasSuffix 函数测试字符串是否有给定的前缀或后缀

```
1 hasPrefix "cat" "catch"
```

双引号(quote)/单引号(squote)括起来

```
1 {{ quote .Values.a }}
2 {{ .Values.a | quote }}
```

cat 函数将多个字符串合并成一个

```
1 cat "hello" "beautiful" "world"
```

indent 开头缩进指定长度

```
1 {{ indent 4 $lots_of_text }}
```

replace 替换

```
1  {{ "I Am Henry VIII" | replace " " "-" }}    -> I-Am-Henry-VIII
```

shuffle 对字符串进行洗牌(打乱字符串)

正则函数:

regexMatch "正则" "值" 匹配到返回true

```
1  regexMatch "^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$" "test@acme.com"
```

regexFind 返回第一个匹配到的值。

```
1  regexFind "[a-zA-Z][1-9]" "abcd1234"
```

regexFindAll 输入字符串匹配正则表达式, 返回所有匹配值

```
1  #最后一个参数表示要返回的子字符串的数量, -1表示返回所有
2  regexFindAll "[2,4,6,8]" "123456789" -1
```

加密函数:

sha1sum "XXX"

sha256sum "XXX"

htpasswd "myUser" "myPassword"

htpasswd 函数使用username 和 password 生成一个密码的bcrypt哈希值。该结果可用于 Apache HTTP Server 的基础认证

时间函数:

now 当前时间

now | date "2006-01-02" 格式化为 Y-M-D



```
now | date_modify "-1.5h"
```

to yaml -> 解析为yaml格式

b64enc/b64dec: 编码或解码 Base64

{{ uuidv4 }} 生成UUID

## 最佳实践

```
1 #嵌套值的每一层都必须检查
2 {{ if .Values.server }}
3   {{ default "none" .Values.server.name }}
4 {{ end }}
```

```
1 #样例
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: {{ template "java-app.fullname" . }}
6   labels:
7     app: {{ template "java-app.name" . }}
8     chart: {{ template "java-app.chart" . }}
9     release: {{ .Release.Name }}
10    heritage: {{ .Release.Service }}
11 spec:
12   {{- if .Values.replicaCount }}
13   replicas: {{ .Values.replicaCount }}
14   {{- end }}
15   revisionHistoryLimit: 5
16   selector:
17     matchLabels:
18       app: {{ template "java-app.name" . }}
19       release: {{ .Release.Name }}
20   template:
```

```
21 metadata:
22   labels:
23     app: {{ template "java-app.name" . }}
24     release: {{ .Release.Name }}
25 spec:
26   containers:
27     - name: {{ .Chart.Name }}
28     #容器镜像应该使用固定的tag或镜像SHA
29     image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
30     imagePullPolicy: {{ .Values.image.pullPolicy }}
31     {{- if .Values.service.internalPort }}
32     ports:
33       - name: http
34         containerPort: {{ .Values.service.internalPort }}
35         protocol: TCP
36     {{- end}}
37     env:
38       - name: POD_NAME
39         valueFrom:
40           fieldRef:
41             fieldPath: metadata.name
42       - name: MY_POD_NAMESPACE
43         valueFrom:
44           fieldRef:
45             fieldPath: metadata.namespace
46       - name: TZ
47         value: {{ .Values.timezone }}
48       - name: JAVA_OPTS
49         value: "-Denv={{ .Values.apollo.apolloEnv }} -Dapollo.meta={{ .V
50       - name: SERVICE_NAME
51         value: {{ .Release.Name }}
52       - name: POD_IP
53         valueFrom:
54           fieldRef:
55             fieldPath: status.podIP
56     {{- range $key,$val := .Values.java }}
57     {{- if $val}}
58     - name: {{ $key | upper }}
59       value: {{ $val }}
60     {{- end}}
61     {{- end }}
62     - name: APOLLO_META
63       value: {{ .Values.apollo.apolloMeta }}
```

```

64     - name: ENV
65       value: {{ .Values.apollo.apolloEnv }}
66     - name: IDC
67       value: {{ .Values.apollo.apolloIDC}}
68   volumeMounts:
69     - name: tz-config
70       mountPath: /etc/localtime
71   {{- if and .Values.livenessProbe.enabled .Values.healthCheck }}
72   livenessProbe:
73     initialDelaySeconds: {{ .Values.livenessProbe.initialDelaySecond
74     timeoutSeconds: {{ .Values.livenessProbe.timeoutSeconds }}
75     failureThreshold: {{ .Values.livenessProbe.failureThreshold }}
76     httpGet:
77       path: {{ .Values.healthCheck }}
78       port: 8080
79   {{- else if and .Values.livenessProbe.enabled .Values.service.inte
80   livenessProbe:
81     tcpSocket:
82       port: {{ .Values.service.internalPort }}
83     initialDelaySeconds: {{ .Values.livenessProbe.initialDelaySecond
84     timeoutSeconds: {{ .Values.livenessProbe.timeoutSeconds }}
85     failureThreshold: {{ .Values.livenessProbe.failureThreshold }}
86   {{- end }}
87   {{- if and .Values.readinessProbe.enabled .Values.healthCheck }}
88   readinessProbe:
89     initialDelaySeconds: {{ .Values.readinessProbe.initialDelaySecon
90     timeoutSeconds: {{ .Values.readinessProbe.timeoutSeconds }}
91     periodSeconds: {{ .Values.readinessProbe.periodSeconds }}
92     httpGet:
93       path: {{ .Values.healthCheck }}
94       port: 8080
95   {{- else if and .Values.readinessProbe.enabled .Values.service.int
96   readinessProbe:
97     tcpSocket:
98       port: {{ .Values.service.internalPort }}
99     initialDelaySeconds: {{ .Values.readinessProbe.initialDelaySecon
100    timeoutSeconds: {{ .Values.readinessProbe.timeoutSeconds }}
101    periodSeconds: {{ .Values.readinessProbe.periodSeconds }}
102   {{- end }}
103
104   {{- if .Values.lifecycle.preStop }}
105   lifecycle:
106     preStop:

```

```

107         exec:
108             command: ["/bin/sh", "-c", "sh /app/nacos_logout.sh"]
109         {{- end}}
110
111     resources:
112 {{ toYaml .Values.resources | indent 12 }}
113     {{- with .Values.nodeSelector }}
114
115
116         nodeSelector:
117 {{ toYaml . | indent 8 }}
118     {{- end }}
119     volumes:
120     - name: tz-config
121       hostPath:
122         path: /usr/share/zoneinfo/{{ .Values.timezone }}
123     {{- if .Values.affinity }}
124     affinity:
125         nodeAffinity:
126             requiredDuringSchedulingIgnoredDuringExecution:
127                 nodeSelectorTerms:
128                 - matchExpressions:
129                     {{- range $key, $val := .Values.affinity }}
130                     - key: {{ $key }}
131                       operator: In
132                       values:
133                         - {{ $val }}
134                     {{- end }}
135             podAntiAffinity:
136                 preferredDuringSchedulingIgnoredDuringExecution:
137                 - weight: 100
138                   podAffinityTerm:
139                       labelSelector:
140                           matchExpressions:
141                           - key: {{ .Values.affinity.dedicated | default "app" }}
142                             operator: In
143                             values:
144                             - {{ template "java-app.fullname" . }}
145                       topologyKey: kubernetes.io/hostname
146                 {{- end}}
147     {{- with .Values.tolerations }}
148     tolerations:
149 {{ toYaml . | indent 8 }}

```

```
150     {{- end }}  
151     imagePullSecrets:  
152     - name: registry-secret  
153
```

