

Variables

See [Helm documentation for full list of available functions](#)

```
name: {{ .Values.storageClassName }}
name: {{ .Values.storageClassName | quote }}
name: {{ .Values.storageClassName | default "default value" }}
name: {{ .Values.storageClassName | required ".storageClassName must be set" }}
name: {{ .Values.storageClassName | trim }}

name: {{ printf "%s-%d" .Values.storageClassName .Values.storageClassVersion }}
name: {{ .Values.storageClassName | replace "{placeholder}" "example" }}

{{ $fullName := printf "%s %s" .Values.firstName .Values.lastName }}

name: {{ .Values.storageClassName | trimAll "/" }}
name: {{ .Values.storageClassName | trimPrefix "/" }}
name: {{ .Values.storageClassName | trimSuffix "/" }}

name: {{ .Values.storageClassName | lower }}
name: {{ .Values.storageClassName | upper }}
```

Built-ins

See [Helm documentation for all options](#)

```
{{ .Release.Name }}
{{ .Release.Namespace }}

{{ .Chart.Name }}
{{ .Chart.Version }}

{{ .Files.Get config.ini }}
```

Conditionals

```
{{ if .Values.enablePersistence }}
# ...
{{ else if .Values.enableFilesystem }}
# ...
{{ else }}
# ...
{{ end }}

# equal, not equal
{{ if eq .Values.environment "production" }}
{{ if ne .Values.environment "production" }}

# and, or
{{ if and (eq .Values.environment "production") (eq .Values.host "minikube") }}
{{ if or (eq .Values.environment "production") (eq .Values.host "minikube") }}
```

```
# not (negation)
{{ if not (eq .Values.environment "production") }}
```

Loops

```
# simple
volumes:
  {{ range .Values.volumeIds }}
  - volumeName: {{ . }}
  {{ end }}

# with named variable
volumes:
  {{ range $volumeId := .Values.volumeIds }}
  - volumeName: {{ $volumeId }}
  {{ end }}

# with index (array) or key (dict)
volumes:
  {{ range $key, $value := .Values.configuration }}
  - {{ $key }}: {{ $value }}
  {{ end }}
```

Indentation

```
env:
  {{ .Values.environmentVariables | toYaml | indent 2 }}

env: {{ .Values.environmentVariables | toYaml | nindent 2 }}
```

Includes

```
# define templates in _helpers.tpl
{{- define "your-project.image" -}}
{{ printf "%s:%s" .Values.image.name .Values.image.tag | quote }}
{{- end -}}

# use in other files
image: {{ include "your-project.image" . }}

# more specific parameters as the scope
{{- define "your-project.someInclude" -}}
{{ . | replace "{placeholder}" "example" }}
{{- end -}}

# usage
foobar: {{ include "your-project.someInclude" .Values.foobar }}
```

Lookup

```
{{ $previous := lookup "v1" "Secret" .Release.Namespace "some-secret" }}
```

```
data:
  {{- if $previous }}
  foobarPassword: {{ $previous.data.foobarPassword | quote }}
  {{- else if .Values.foobarPassword }}
  foobarPassword: {{ .Values.foobarPassword | b64enc | quote }}
  {{- else }}
  foobarPassword: {{ randAlphaNum 40 | b64enc | quote }}
  {{- end }}
```

Fail

```
{{ if eq .Values.storageClassName "foobar1" }}
# ...
{{ else if eq .Values.storageClassName "foobar2" }}
# ...
{{ else }}
  {{ fail ".storageClassName is not recognized" }}
{{ end }}
```

Dates

See [Helm documentation](#) for notes about formatting

```
# ISO 8601, format string is provided as a lookalike-string
{{ now | date "2006-01-02T15:04:05" }}
```

Base64

```
{{ .Values.someData | b64enc }}
{{ .Values.someData | b64dec }}
```

UUIDs

```
id: {{ uuidv4 }}
```

Crypto

```
{{ .Values.someData | sha256sum }}

{{ .Values.someData | encryptAES "secret key" }}
{{ .Values.someData | decryptAES "secret key" }}
```

Source code for this page can be found on [GitHub](#).