Helm Pipelines

- A pipeline is a sequence of commands, functions and variables chained together

```
character: {{ .Values.character | default "Learning" | quote }}
```

Template Functions

- Within actions and pipelines, there are template functions which we can use.
- Functions provide a means to transform the data
- Most of the functions provided by helm are designed to be useful when generating charts.
- The functions range from simple like indent and nindent functions to indent output to complex ones that are able to reach into cluster and get information on current resources and resource types.
- Note: Most of the functions found in helm template are provided by a library named Sprig ref: http://masterminds.github.io/sprig/
- Helm templates have more than hundred function ref: https://helm.sh/docs/chart_template_guide/function_list/

Methods

- Helm also includes functions to detect the capabilities of K8s cluster and methods to work with files
- The .Capabilities object has the method .Capabilities.APIVersions.Has which takes a single argument for the K8s AP or type we want to check the existence of
- The other place where we can find methods is on .Files. It includes the following methods
  - .Files.Get name: Gets the content of the file name
  - .Files.GetBytes
  - .Files.Glob
  - .Files.AsConfig: Takes the file group and returns a flattened YAML suitable to include in the data section of K8s ConfigMap
  - .Files.AsSecret
  - .Files.Lines

Flow Control

- Go templates have if and else statements along with something similar but slightly different called with. if and else
- Lets assume in values.yaml file we have a section on ingress with enabled property

```
ingress:
  enabled: false
```

- In the ingress.yaml that creates the ingress resource for K8s the first and last lines are for the if statement

```
{{- if .Values.ingress.enabled  -}}
...
{{- end }}
```

Loops

- A sample list in yaml

```
characters:
  - ironman
  - thor
  - hulk

movies:
  avengers: 'This is good movie'
  wintersoldier: 'This is too good movie'
```

- We can generate above mentioned lists by using range

```
characters:
{{- range .Values.characters }}
  - {{ . | quote }}
{{- end }}
```

- We can generate a dictionary or map

```
movies:
{{- range $key, $value := .Values.movies }}
  - {{ $key }}: {{ $value | quote }}
```

Experiment- Creating a simple manifest for ngnix

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-example
spec:
  containers:
  - name: ubuntu
    image: ubuntu:trusty
    command: ["sleep"]
    args: ["1d"]
```

```
apiVersion: v1
kind: Pod
metadata:
  name: {{ .Values.pod.name }}
spec:
  containers:
  - name: ubuntu
    image: {{ .Values.image.repository | default "alpine" }}:{{ .Values.image.tag | default "latest" }}
    command: {{ .Values.container.command }}
    args: {{ .Values.container.args }}
```

The values

```yaml
# Default values for inventory.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

pod:
  name: "experiment-1"

image:
  repository: ubuntu
  pullPolicy: IfNotPresent
  # Overrides the image tag whose default is the chart appVersion.
  tag: "latest"

container:
  command: ["sleep"]
  args: ["1d"]
```