**Helm**



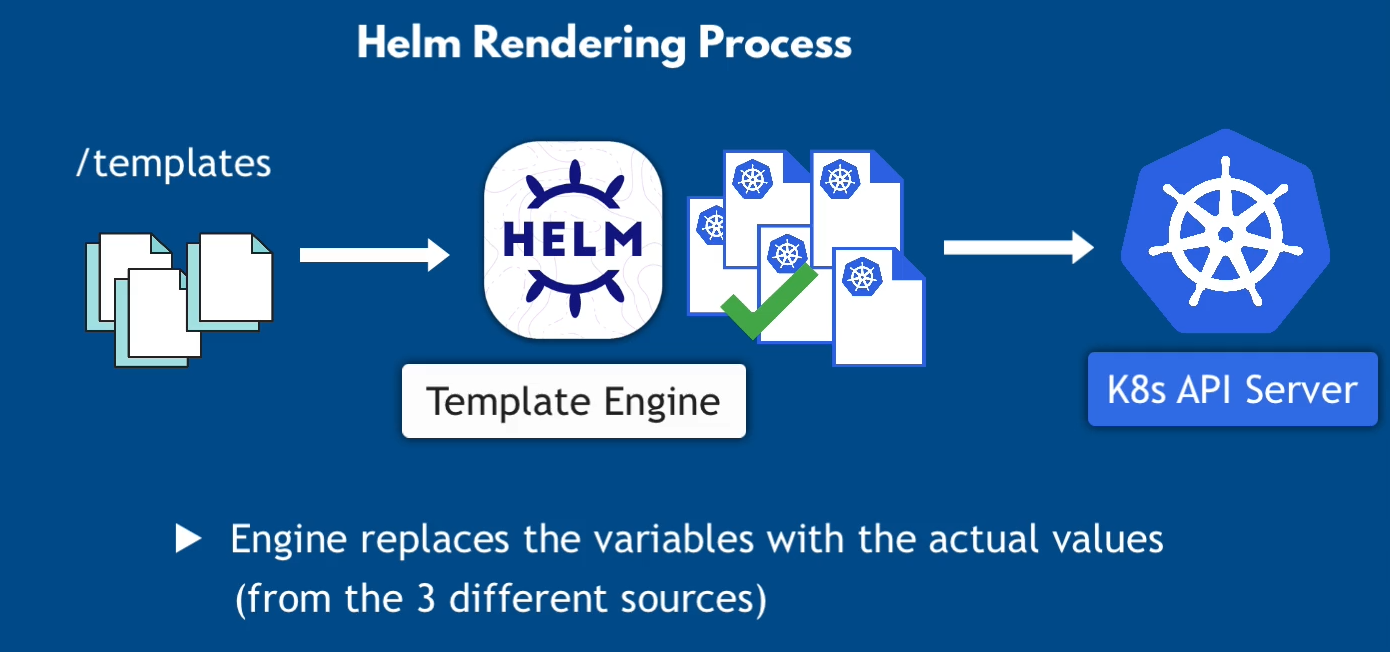
Helm is a tool for managing Charts. Charts are packages of pre-configured Kubernetes resources.

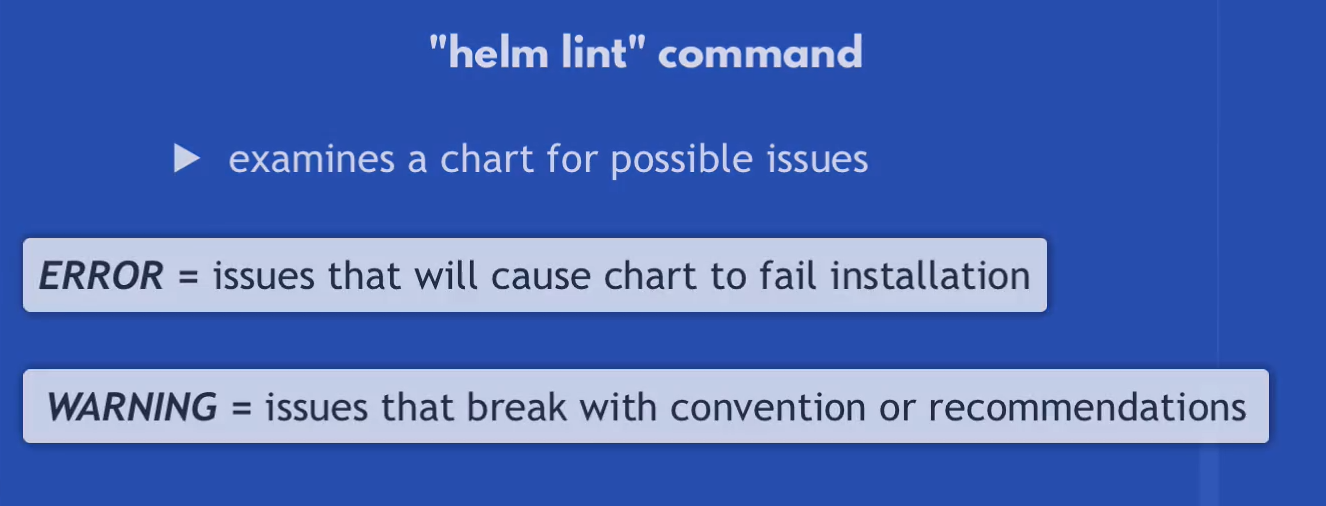
Use Helm to:

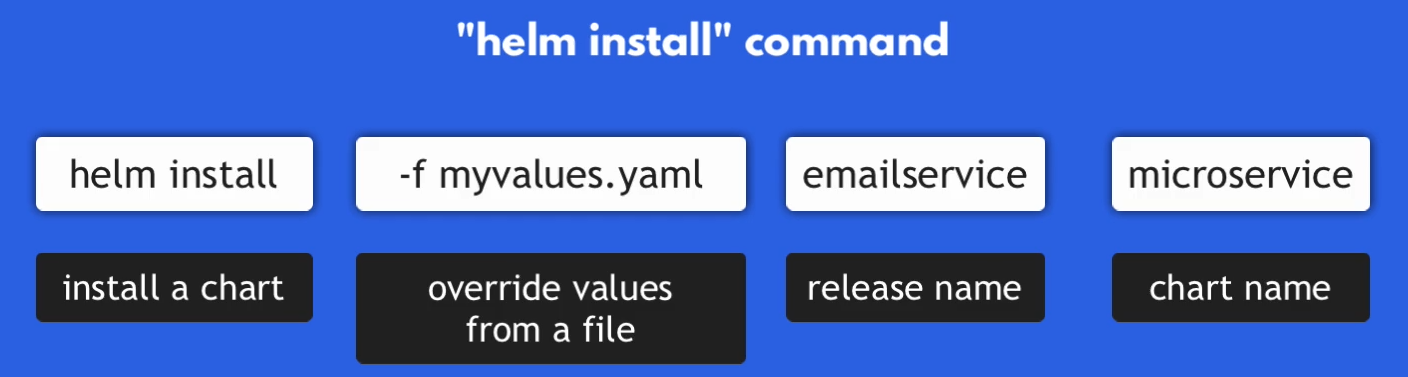
* Find and use popular software packaged as Helm Charts to run in Kubernetes
* Share your own applications as Helm Charts
* Create reproducible builds of your Kubernetes applications
* Intelligently manage your Kubernetes manifest files
* Manage releases of Helm packages

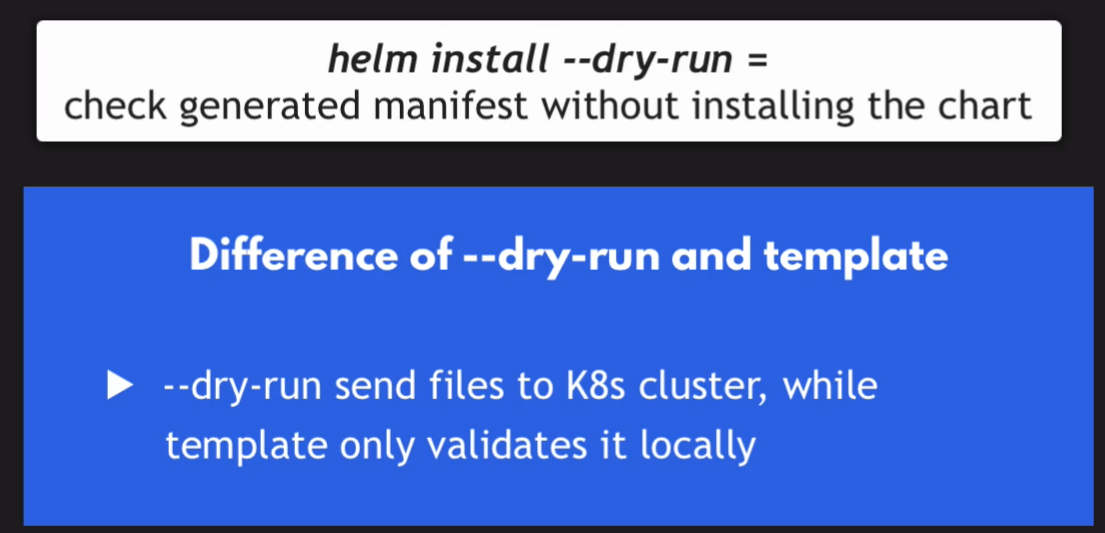
What does helm do:

* Helm renders your templates and communicates with the Kubernetes API
* Helm runs on your laptop, CI/CD, or wherever you want it to run.
* Charts are Helm packages that contain at least two things:
* A description of the package (Chart.yaml)
* One or more templates, which contain Kubernetes manifest files
* Charts can be stored on disk, or fetched from remote chart repositories (like Debian or RedHat packages)

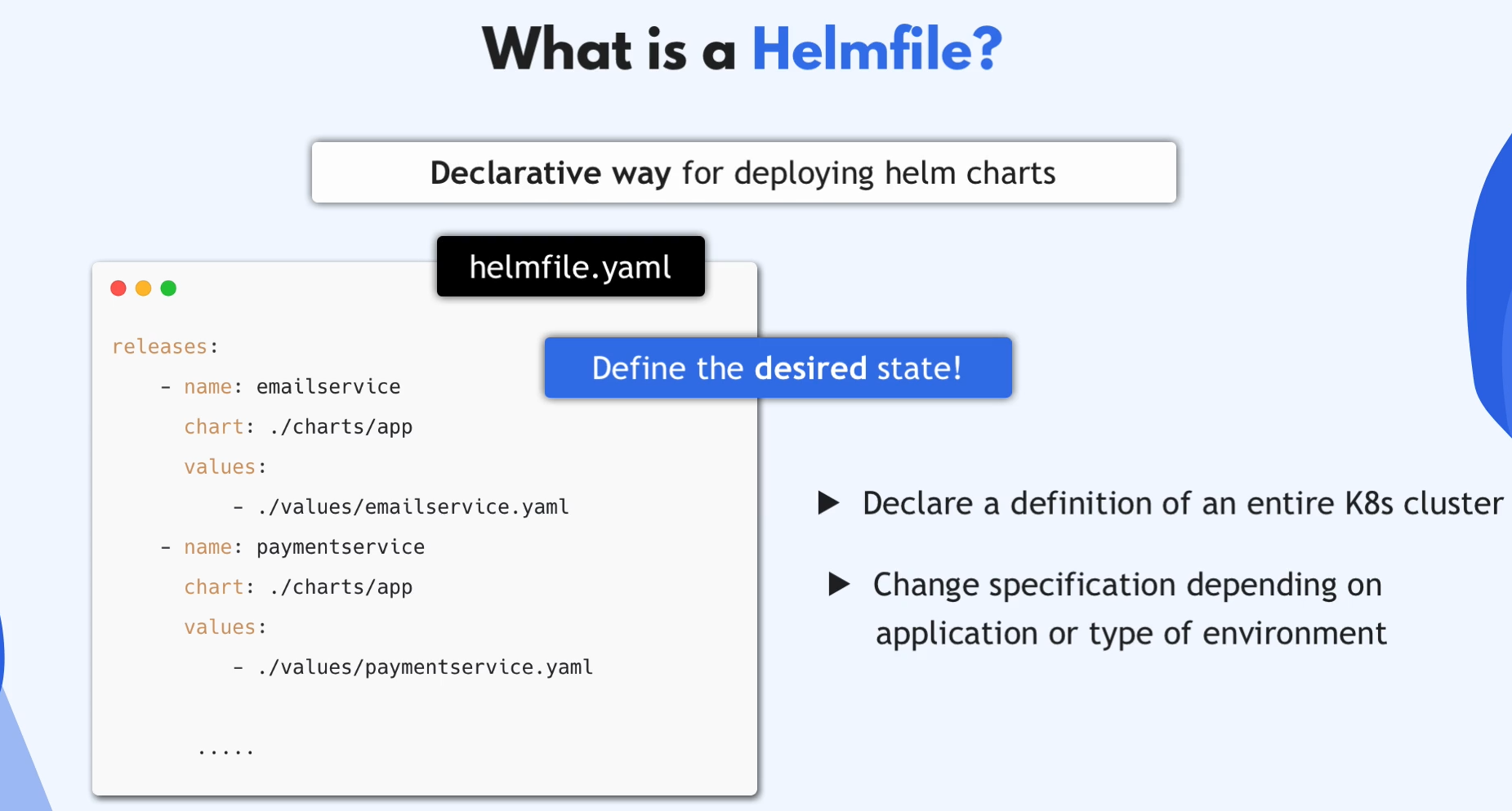






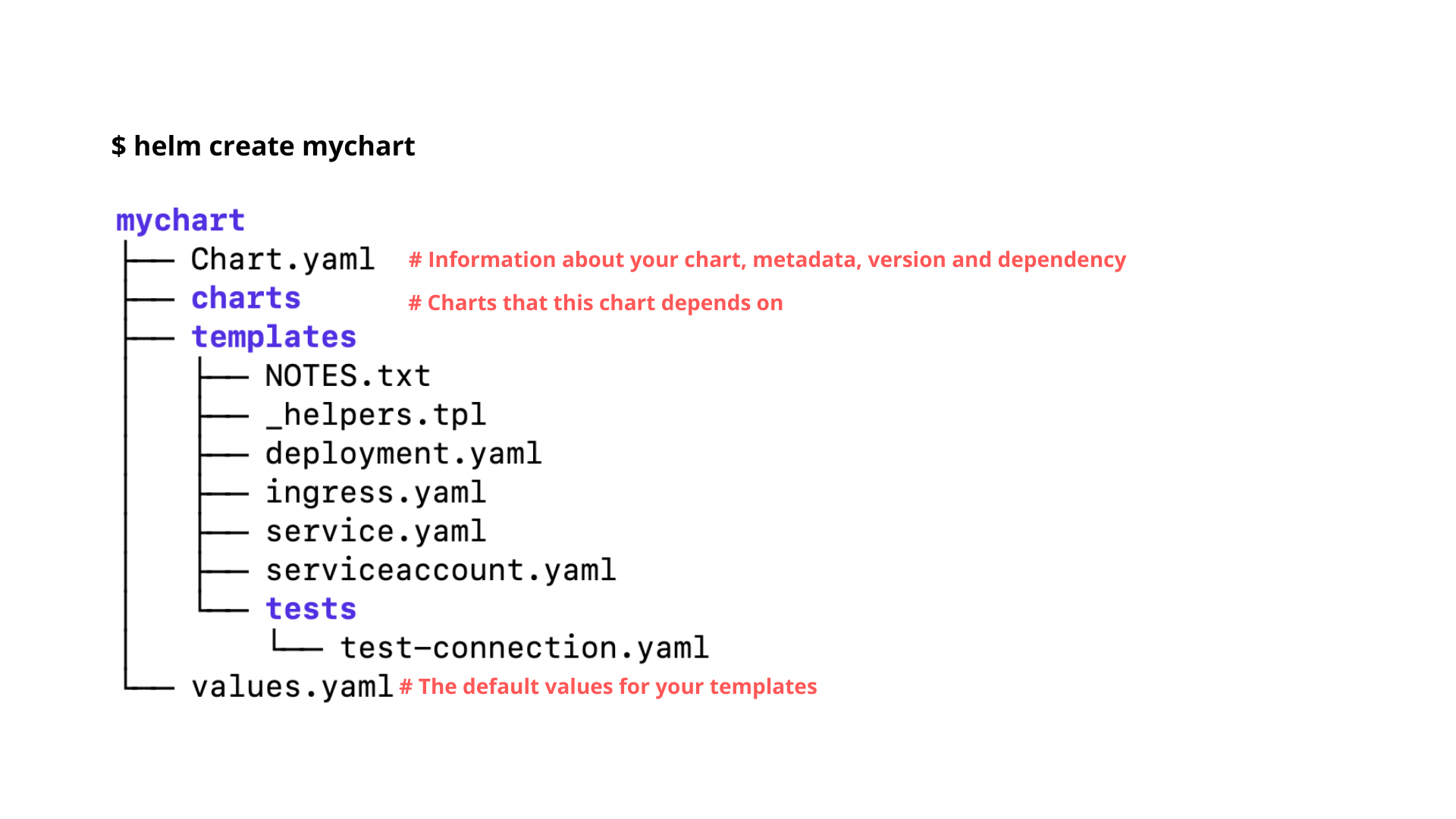


**Helm File**



How to Create a Helm Chart

It's pretty easy to create a chart in Helm. First, you need to have Helm installed. Then, just type in helm create <chart name> and it will create a directory filled with files and other directories. Those files are required for Helm to create a chart.

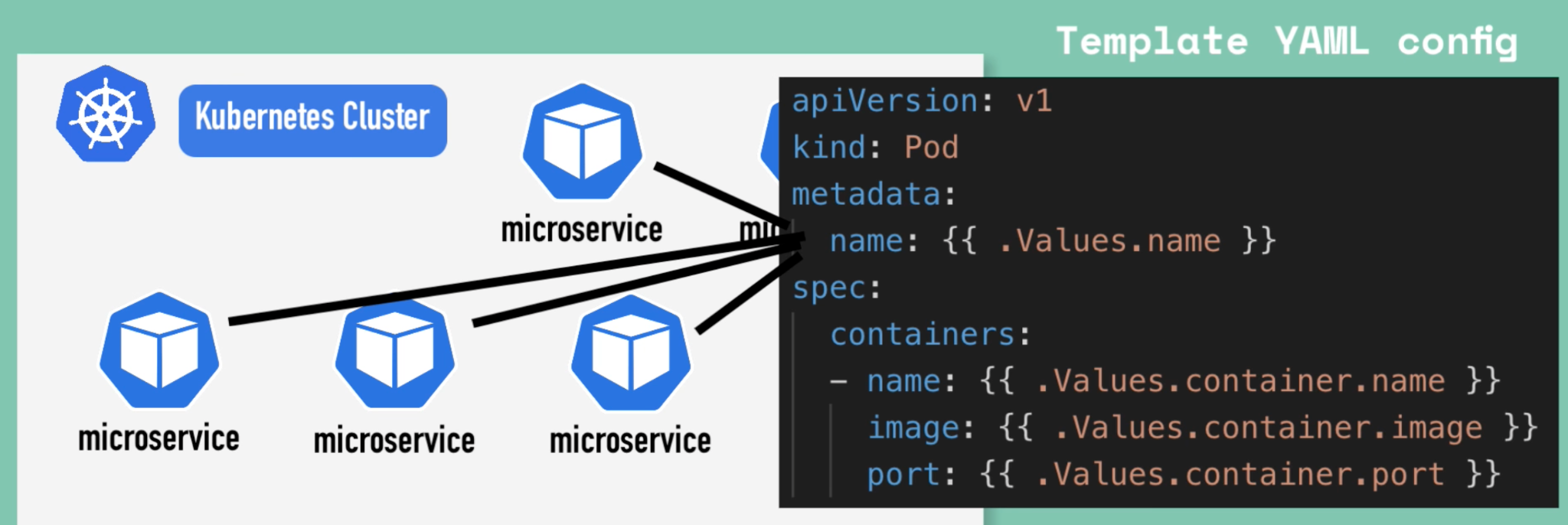


**chart.yaml**: This is where you'll put the information related to your chart. That includes the chart version, name, and description so you can find it if you publish it on an open repository. Also in this file you'll be able to set external dependencies using the dependencies key.

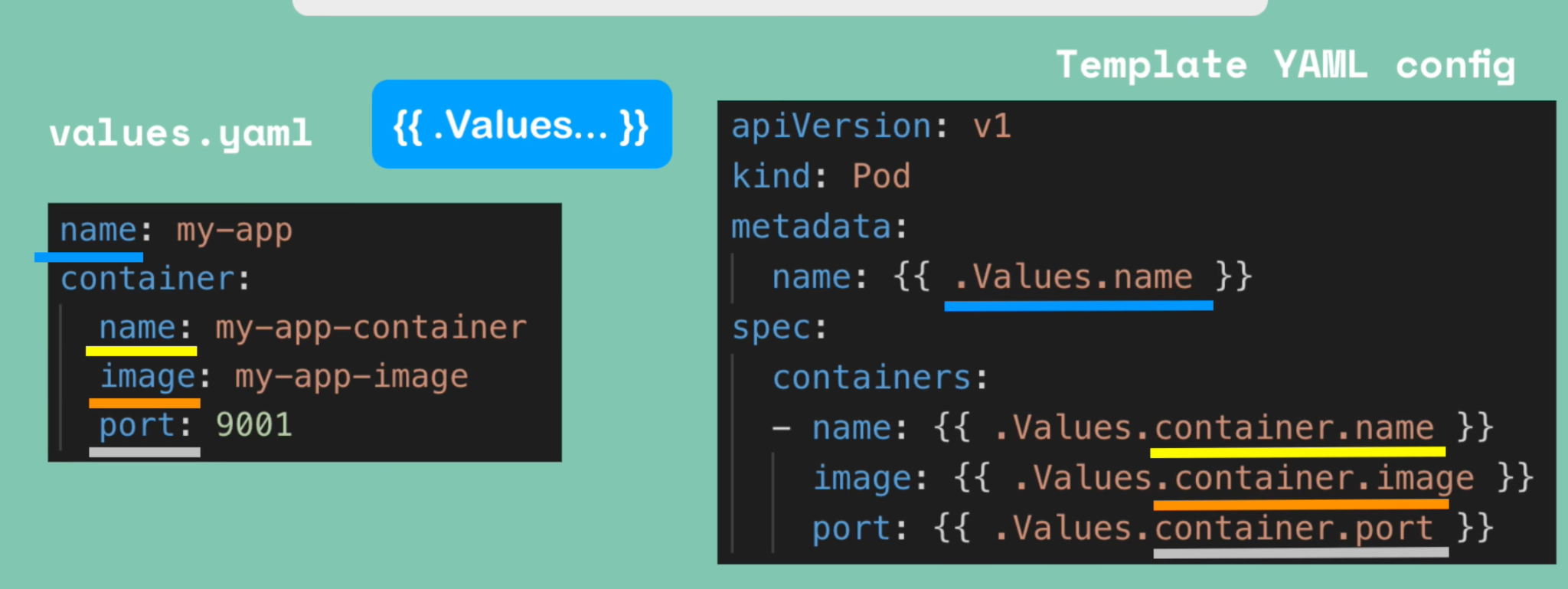
**values.yaml**: Like we saw before, this is the file that contains defaults for variables.

templates (dir): This is the place where you'll put all your manifest files. Everything in here will be passed on and created in Kubernetes.

**charts**: If your chart depends on another chart you own, or if you don't want to rely on Helm's default library (the default registry where Helm pull charts from), you can bring this same structure inside this directory. Chart dependencies are installed from the bottom to the top, which means if chart A depends on chart B, and B depends on C, the installation order will be C ->B ->A.

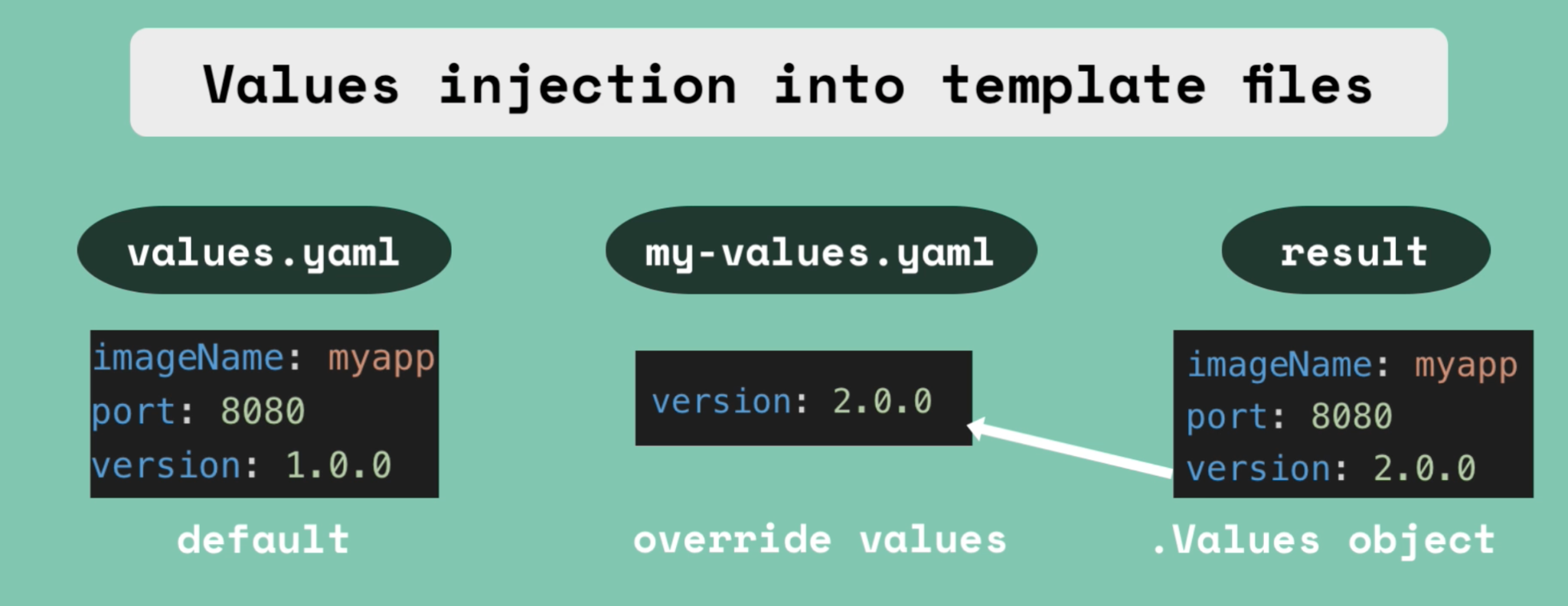


{{ .values… }}



Objects which are created based on the values defined.

Values are either defined via yaml file or with --set flag



dbie-gke is our chart, inside it we have

templates => this folder contains all the original kubernetes files that are turned into helm templates, they are written in GoLang templating language (similar to Jinja)

Chart.yaml => this file contains the information about this chart

values.yaml => contains default values for this chart, values will be injected to the template file

outside the chart we have:

values folder => inside it we have values file for each instance of the chart, these will replace the variables of values.yaml file that is inside the chart.

helmfile.yml => here we define each instance of the chart and where it's values are located.

**useful Helm commands**

generates all base files needed for a helm chart, microservice can be changed to any name for the chart

helm create microservice

this will inject the values into the chart to make sure that it works correctly

helm template -f values/test-values.yml microservice

checks the yaml file's syntaxt for both values file and helm template files

helm lint -f values/test-values.yml microservice

install the chart into the cluster, this should be done for each microserivce that wants to be added into the cluster based on the chart

helm install -f values/test-values.yml CutomName microservice

same as above, but this one injects values to kubernetes cluster temporarily to make sure it works fine

helm install --dry-run -f values/test-values.yml microservice

shows all microservices running

helm ls

unistall helm services

helm uninstall CutomName

install helmfile tool, this is a plugin for Helm, allows installing several instances of the chart at once

brew install helmfile

deploy helm charts via helmfile

helmfile sync

shows all the applied charts

helmfile list

delete all the service

helmfile destroy

Our projects link: <https://gitlab.com/l1905/multi-cloud-operations-and-sre/db-engineering/dbie-gke/-/tree/helm?ref_type=heads>

Useful links:

<https://helm.sh/>

<https://helm.sh/docs/>

<https://artifacthub.io/>