Umbrella Helm Chart

الـ HelmChart هو عبارة عن Package فيها ملفات بتقول لـ Kubernetes اعمل الـ Kubernetes resources بالطريقة دي

myapp/
- Chart.Yaml
- Values.Yaml
- templates
  - deployment.yaml
  - service.Yaml
  - ingress.Yaml

~~لما تعمل~~ helm install myapp ./my

الـ Helm بياخد الـ Templates + Values.Yml ويحولهم الى Kubernetes resources

طب ايه هو الـ Umbrella Chart

(Parent Chart) umbrella Chart = Helm Chart

وظيفته الأساسية إنه يجمع Charts تانية تحته

يعني بدل ما يكون عندي

هعمل
- AppChart
- DB Chart
- RabbitMQ Chart

الحل

Install هعمل لكل واحدة لوحدها

helm install app ./app

helm install DB ./DB

VProfile
- Chart.Yaml
- Values.Yaml
- Charts
  - app
  - DB
  - MQ

- الـ VProfile هنا هو الـ umbrella Chart

الحل: ← اعمل Chart واحدة رئيسية

Umbrella Chart

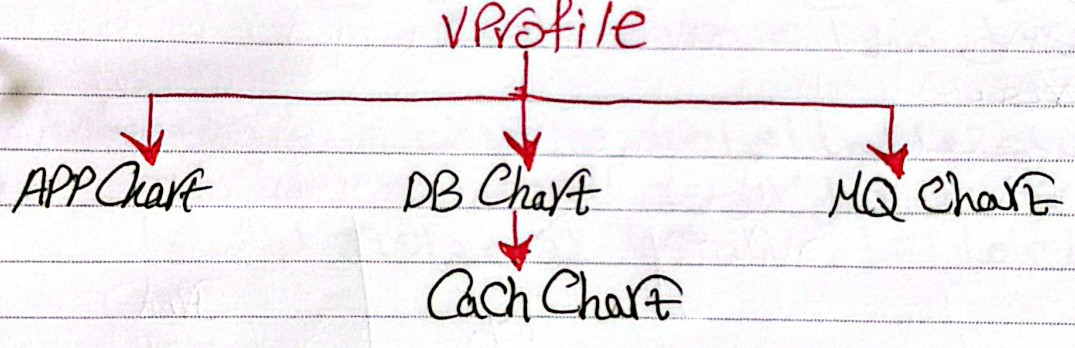
تخيل عندك مشروع كامل ← VProfile

- APP
- DB
- Mem
- Rabbit

كل Component ممكن يكون ليه Helm Chart الخاص بيه

- app-Chart
- db-Chart
- memcached-Chart
- rabbitmq-Chart

ال Umbrella Chart بتجمعهم فوقهم



Umbrella Chart

ال umbrella Chart بتعرف ال Charts
اللي تحتها من Chart.Yaml.
بقول لل Helm ال Chart بتاعة ال VProfile
عندها Dependencies اسمها MQ, app, DB

يعني ال VProfile معتمد على MQ, app, DB

طيب لما اجي اعمل Install

* helm install vprofile ./vprofile
ساعتها ال Helm يشوف ال Dependencies
هيجمع مجموع ال Charts دي مع ال Parent
ويعمل منها Release واحدة

* وساعتها لما اعمل helm upgrade --install vprofile ./vp
هكون بعمل deploy للـ application stack كله
بدل ما اعمل Install لكل واحدة لوحدها

Umbrella Chart Values.yaml

هنا الـ umbrella Chart قوية جداً

```
app:
  replicaCount: 3
db:
  persistence
    size: 10Gi
rabbitmq:
  replicaCount: 1
```

مثلاً

الـ Parent بتتحكم
في الـ Configuration
الخاصة بالـ Child Charts

وده بيسمح لي
اعمل Configuration
مركزية للـ Stack كله

Umbrella Chart

مش شرط نختار دايماً ال Umbrella Chart

لأن ال Umbrella Chart مش نسخة احسن من ال Helm Chart

السؤال هنا مش هل ال Umbrella أحسن؟

السؤال الصح ← هل عندي مجموعة Helm Chart مستقلة محتاج أجمعها وأديرها مع بعض

لو الإجابة لا ← يبقى مفيش سبب قوي نستخدم فيه ال Umbrella

Helm Chart Umbrella

نطبق دا على مشروع الـ VProfile

VProfile
- app
- db
- rabbitmq
- memcached
- web

ممكن تقول عندي 5 Components بالتالي انا هعمل 5 Chart واحطهم تحت umbrella

لكن هنا المشكلة

كل الـ 5 Components حول فعلاً 5 APP مستقلة

فعلياً لا لأنهم اجزاء من نفس Application Stack

بالتالي انا مش بقول عندي 5 Products

انا بقول عندي APP واحد مكون من عدة Kubernetes Resources

طيب أي المشكلة لو عملنا umbrella

فعلياً هو مفيش خطأ تقني

لكن مكنش ليها قيمة معمارية كافية الـ APP

السبب الأول ← مفيش استقلال حقيقي بين الـ Components

VProfile
- Chart.yaml
- Charts
  - app
  - cach
  - db
  - MQ

هل هـ ممكن يكتب helm install db ./charts/db

ويشغل الـ DB لوحده ؟ هو اكيد ممكن

لكن ده مش الـ useful في الـ Arch بتاعي

لأن الـ DB جزء من VProfile deployment

Helm-Umbrella Chart

السبب التاني: هزود ال Complexity انا عندي

```
vProfile
├ Chart.yaml
├ Values.yaml
└ templates
  ├ app.yaml
  ├ db.yaml
  ├ rmq.yaml
  ├ mc.yaml
  └ web.yaml
```

لما أعمل helm upgrade --install vProf
لحد هنا الدنيا بسيطة

لو عملت umbrella

```
vProfile
├ Chart.yaml
├ Values.yaml
└ Charts
  ├ app/
  │ ├ Chart.yaml
  │ ├ Values.yaml
  │ └ Templates
  │   └ deployment.yaml
  └ db/
    ├ Chart.yaml
    ├ Values.yaml
    └ Templates/
      └ deployment.yaml
```

مع ال Helm Chart
Chart 1
↓
Templates 5

مع ال Umbrella Chart
1 Parent Chart
5 Child Charts
↓
كل Child Chart عنده Template بتاعته

دا هو ال Complexity اللي قصدي عليه

Helm Umbrella Chart

السبب التالت: ← ال Values هتبقى كده

دلوقتي ← image:
tag: "1.1"

لكن مع ال Umbrella

```
app:
  image:
    tag: "1.1"
db:
  image:
    tag: "1.1"
```

طبعاً ممكن استخدم global:

```
imageTag: "1.1"
```

لكن انت أضفت Convention جديدة لإدارة ال Values

← مش مشكلة لو عملت كدا لكن لو المشروع مش محتاج كل دا ليه اعمله ؟

Helm Umbrella Chart

السبب الرابع: ال umbrella بيبقى قوي
لما يكون عندك Dependencies مستقلة
AWS load Balancer Controller
Prometheus
Grafana
Vault
Velero

هنا كل واحد
- ليه Chart خاص
- ليه Version
- ليه Configuration
- ممكن يتحدث independently
- مصدره مختلف
- وليه lifeCycle مختلف

هنا ال Umbrella بقى ليه قيمة حقيقية

انا كنت أقدر أستخدم umbrella Chart لكن اخترت
Single APP Chart لأن ال APP وال DB
وال MQ وال Cached وال Web في المشروع
بتاعي يعتبروا Components لنفس ال APP Stack
ومش HelmApp مستقلة محتاجة lifeCycle منفصل
استخدم SubChart كان هيضيف abstraction
و Complexity في ادارة ال values وال dependencies
بدون benefit حقيقي من ناحية ال reuse
أو ال Independent deploy لكن في ال umbrella Chart
ال Platform layer هيكون مناسب جداً لأن عندي
AWS load Balancer Controller زي independent third-Party Charts
و Prometheus و Vault و Velero