Rancher 2.1 Catalog Apps and Helm Charts

RANCHER LABS

目录

1. Brief Intro of Helm and Rancher Catalogs
2. Rancher Catalog 2.1 vs Helm
3. How to Create and Add Custom Catalogs
4. What is Operator and How it works it Rancher

Brief Intro of Helm & Catalog Apps

- A package manager for Kubernetes
- Similar to the docker-compose in Rancher 1.6 but in k8s manner
- Helps to build, install and upgrade even the most complex k8s apps (e.g Redis, Memcached, MongoDB, RabbitMQ)

Helm: Core Concepts

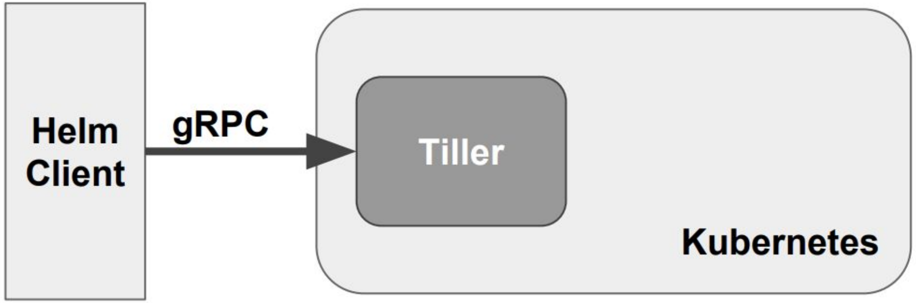
- Chart - expert build recipe for installing an application
- Values - user supplied configuration
- Release - instance of Chart + Values that get deployed in K8S

Rancher Catalog: Core Concepts

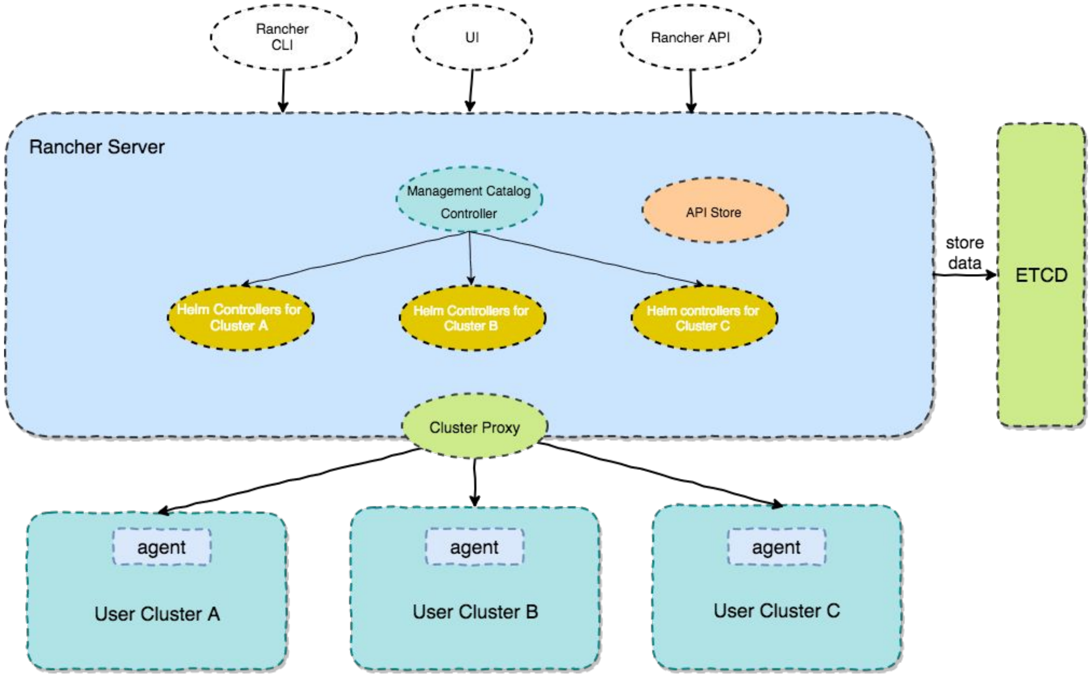
- Chart - expert build recipe for installing an application
- Values - user supplied configuration
- Release - instance of Chart + Values that get deployed in K8S
- questions.yml - simply user inputs
- app-reame.md - add brief intro of the catalog app
- Multi-tenancy support and persistent data

Helm: Architecture

1. Server-side component
2. Runs as a pod in the cluster
3. Manages releases in your cluster
4. No multi-tenancy support (one RBAC per tiller per namespace)
5. No real persistent data



Rancher Catalog: Architecture

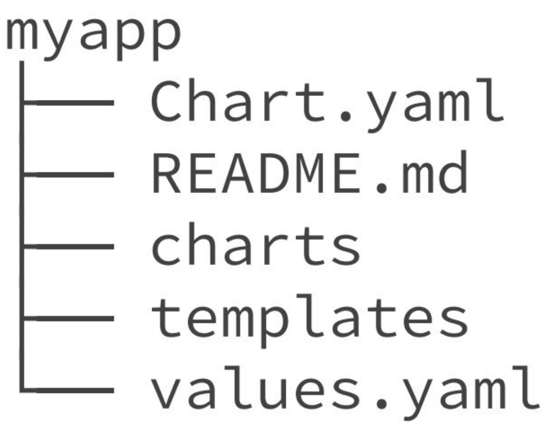


1. Persistent data - catalog controllers stores charts and apps as CRDs resources
2. One Cluster per helm controller for centralized RBAC
   a. User Impersonation(RBAC)
   b. Allows run multiple tiller server at the same controller

How to Create & Add Custom Catalogs

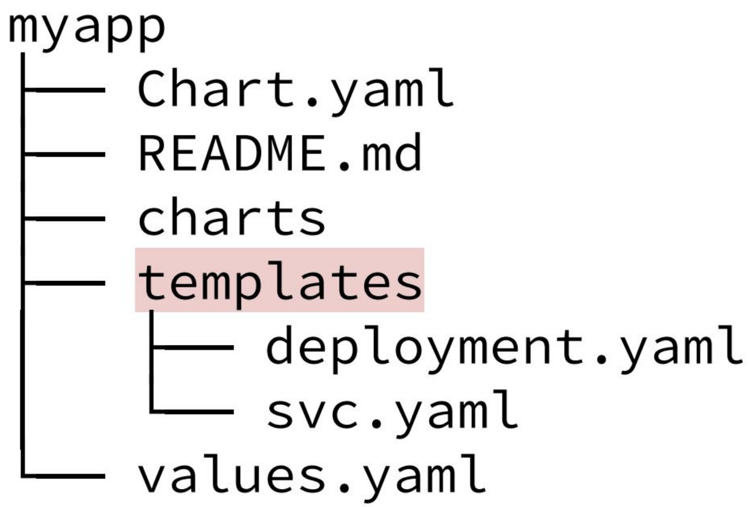
1. Fork and Modify from Helm Libraries (https://github.com/helm/charts)
2. Use Helm to Create New Chart (refer to docker-compose or docker file)
3. Adding Rancher question.yml and app-readme.md file
4. Push to your chart to Git or Https Helm Repo

Helm: Chart Structure



```
myapp
├── Chart.yaml
├── README.md
├── charts
├── templates
└── values.yaml
```

Helm: Templates



```
myapp
├── Chart.yaml
├── README.md
├── charts
├── templates
│   ├── deployment.yaml
│   └── svc.yaml
└── values.yaml
```

Helm: Configurations

```
myapp
├── Chart.yaml
├── README.md
├── charts
├── templates
├── values.yaml
└── requirements.yaml
```

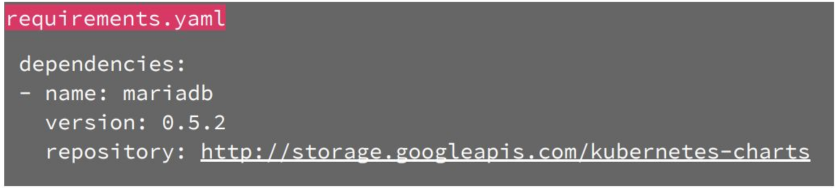
```
values.yaml

image: mycompany/myapp:1.0.0
imagePullPolicy: IfNotPresent
service:
  port: 80
```

```
templates/deployment.yaml
apiVersion: extensions/v1beta1
kind: Deployment
spec:
  template:
    spec:
      containers:
      - name: {{ .Chart.Name }}
        image: "{{ .Values.image }}"
        imagePullPolicy: {{ .Values.imagePullPolicy }}
        ports:
        - containerPort: {{ .Values.service.port }}
```

Helm: Dependencies

```
myapp
├── Chart.yaml
├── README.md
├── charts
├── templates
└── values.yaml
└── requirements.yaml
```



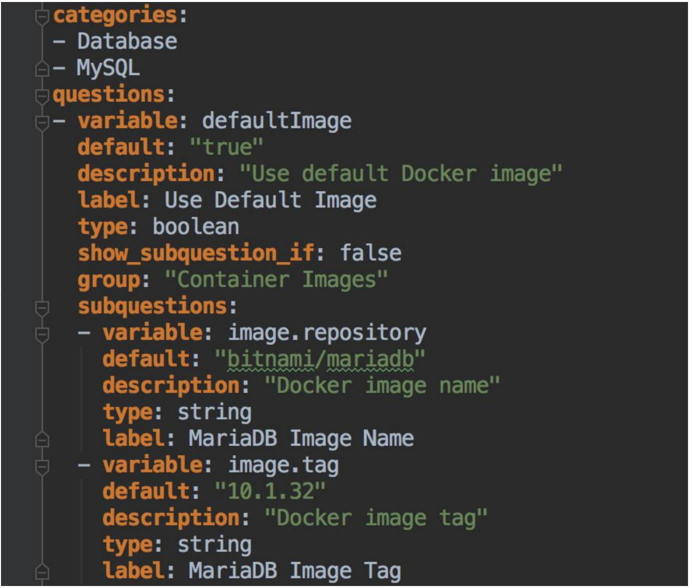
requirements.yaml

```
dependencies:
- name: mariadb
  version: 0.5.2
  repository: http://storage.googleapis.com/kubernetes-charts
```

Rancher Catalogs: UI Enhancement

```
myapp
├── Chart.yaml
├── README.md
├── charts
├── templates
└── values.yaml
└── requirements.yaml
-----  questions.yml

-----  app-readme.md
```



```yaml
categories:
- Database
- MySQL
questions:
- variable: defaultImage
  default: "true"
  description: "Use default Docker image"
  label: Use Default Image
  type: boolean
  show_subquestion_if: false
  group: "Container Images"
  subquestions:
  - variable: image.repository
    default: "bitnami/mariadb"
    description: "Docker image name"
    type: string
    label: MariaDB Image Name
  - variable: image.tag
    default: "10.1.32"
    description: "Docker image tag"
    type: string
    label: MariaDB Image Tag
```

Rancher Catalogs: Features

1. Rancher 2.1 bring tiller back - allows better CRD integration and lifecycle-hooks
2. Centralized RBAC for user management
3. Rancher provided catalog apps detailed page
4. Rancher natively supports custom CRD resources via helm and CLI

Supported Helm Repo Servers

1. Git compatible server - Github, Bitbucket and Gitlab etc.,
2. Http/s Server (ChartMusem, Jfrog Helm Library, nginx server etc.,)

Recap:

1. Fork and Modify from Helm Libraries (https://github.com/helm/charts)
2. Use Helm to Create New Chart (refer to docker-compose or docker file)
3. Adding Rancher question.yml and app-readme.md file
4. Push to your chart to Git or Https Helm Repo

What is Operator & How it works with Rancher

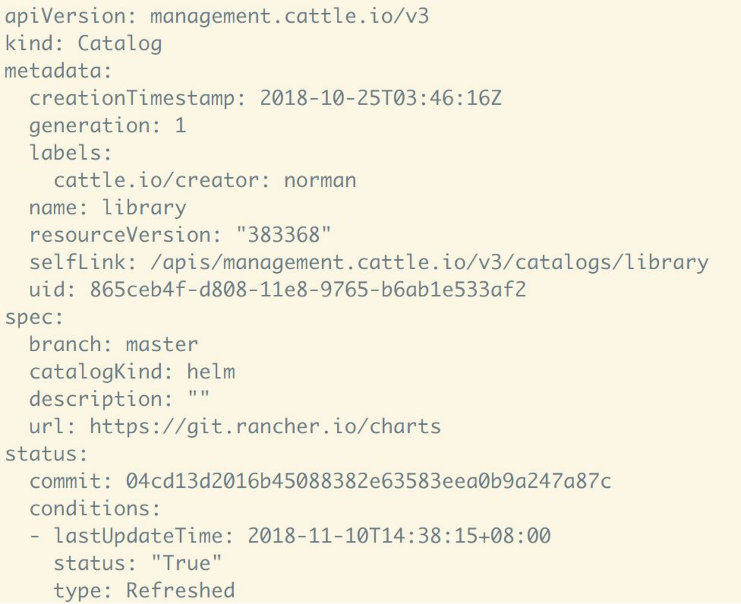
1. Operator cores: CRD and k8s Controllers

2. Controller Tools:

   a. Kubebuilder
   https://github.com/kubernetes-sigs/kubebuilder

   b. Kubernetes Sample Controller
   https://github.com/kubernetes/sample-controller

   c. Go-Skel (Rancher)
   https://github.com/rancher/go-skel



```
apiVersion: management.cattle.io/v3
kind: Catalog
metadata:
  creationTimestamp: 2018-10-25T03:46:16Z
  generation: 1
  labels:
    cattle.io/creator: norman
  name: library
  resourceVersion: "383368"
  selfLink: /apis/management.cattle.io/v3/catalogs/library
  uid: 865ceb4f-d808-11e8-9765-b6ab1e533af2
spec:
  branch: master
  catalogKind: helm
  description: ""
  url: https://git.rancher.io/charts
status:
  commit: 04cd13d2016b45088382e63583eea0b9a247a87c
  conditions:
  - lastUpdateTime: 2018-11-10T14:38:15+08:00
    status: "True"
    type: Refreshed
```

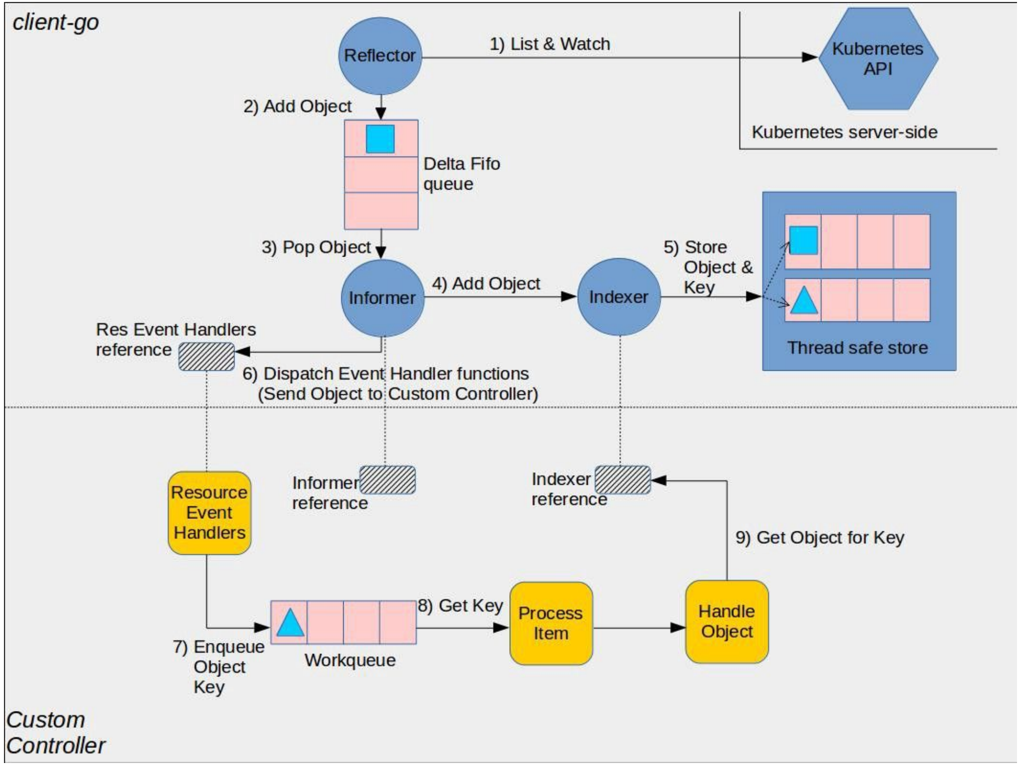
What is Operator

Informer reference:

This is the reference to the Informer instance

Indexer reference:

This is the reference to the Indexer instance



LIVE DEMO

serenataflowers.com
"Thank You"