Guidelines for Labels and Annotations for OpenShift applications

Version 0.9, 07.01.2019

Table of Contents

Terminology

Software System

Highest level of abstraction that delivers value to its users, whether they are human or not. An application is composed of one or more `Component`s.

Application

An abstraction that encapsulates the value delivered by the software system. A software system is composed of one or more `Application`s.

Component

A component is a set of kubernetes resources, for hosting code or data, that needs to be running in order for the overall software system to work. Each component is a seperately deployable and runnable unit.

Labels

The following table defines common labels applications deployed on OpenShift should use Labels that are marked *REC* are recommended. Those marked *OPT* are optional. The labels should not be relied upon for operational purposes.

You can find more information on the Kubernetes labels, prefixed with app.kubernetes.io, in the Kubernetes documentation.

TIP	Labels must be applied to ALL resources, DeploymentConfigs, BuildConfigs, Services, Routes, ConfigMaps, PersistenVolumeClaims etc., including any custom resource definitions (CRDs).

Table 1. Labels

Name	Status	Description	Example
app.kubernetes.io/part-of	REC	The name of the top level software system this resource is part of	ticketmonster
app.kubernetes.io/name	REC	The name, reflecting component.	mysql
app.kubernetes.io/component	REC	This is the role/type of the component.	frontend
app.kubernetes.io/managed-by	REC	The tool being used to manage the operation of the component	odo
app.kubernetes.io/instance	OPT	A name identifying the application, usually used if different from `app.kubernetes.io/name`	accounts
app.kubernetes.io/version	OPT	The current version of the component (e.g., a semantic version, revision hash, etc.)	1.0.0

Table 2. Values for `app.kubernetes.io/component` label

Value	Description
frontend	Serves the UI or part of the UI for an application
backend	Usually an application code that is running on a runtime or framework.
database	Data persistence
integration	Integration middleware such as API gateways or single-sign-on software
cache	
queue	

Annotations

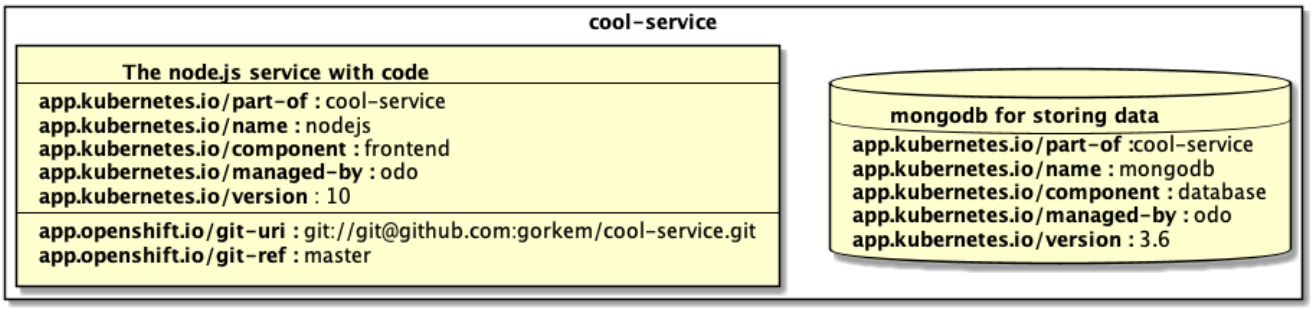
Table 3. Annotations

Name	Status	Description	Example
app.openshift.io/git-uri	REC	The git url for the source code of application	git://git.kernel.org/pub/scm/linux.git
app.openshift.io/git-ref	REC	Branch, tag or commit hash for the application if omitted master is assumed	v1.0.0

Examples

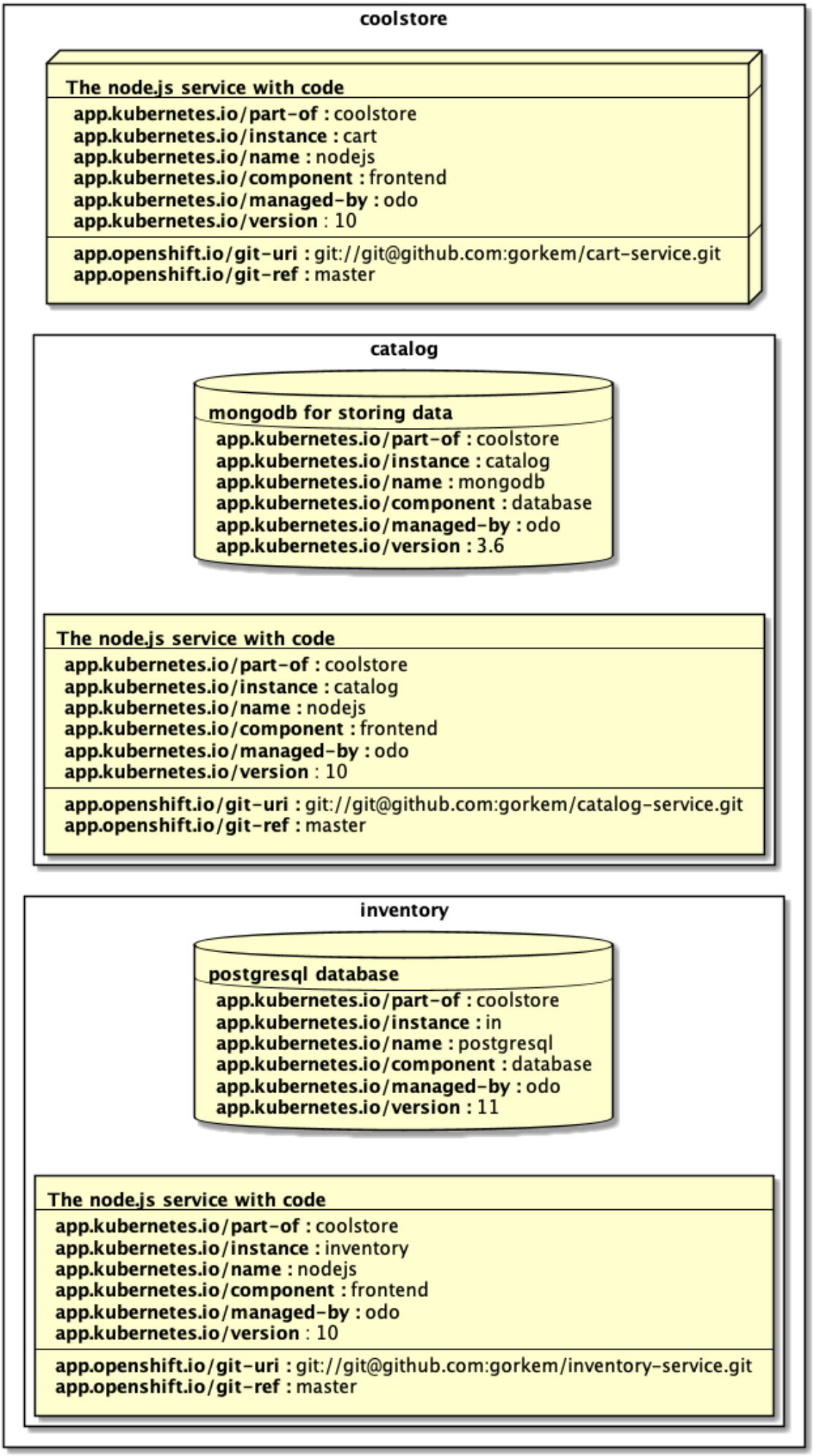
Simple microservice with a database

Node.js based rest service with a database (mongodb) backing



A complex system with multiple services

TODO: Complete the example


coolstore
The node.js service with code
app.kubernetes.io/part-of : coolstore
app.kubernetes.io/instance : cart
app.kubernetes.io/name : nodejs
app.kubernetes.io/component : frontend
app.kubernetes.io/managed-by : odo
app.kubernetes.io/version : 10
app.openshift.io/git-uri : git://git@github.com:gorkem/cart-service.git
app.openshift.io/git-ref : master
catalog
mongodb for storing data
app.kubernetes.io/part-of : coolstore
app.kubernetes.io/instance : catalog
app.kubernetes.io/name : mongodb
app.kubernetes.io/component : database
app.kubernetes.io/managed-by : odo
app.kubernetes.io/version : 3.6
The node.js service with code
app.kubernetes.io/part-of : coolstore
app.kubernetes.io/instance : catalog
app.kubernetes.io/name : nodejs
app.kubernetes.io/component : frontend
app.kubernetes.io/managed-by : odo
app.kubernetes.io/version : 10
app.openshift.io/git-uri : git://git@github.com:gorkem/catalog-service.git
app.openshift.io/git-ref : master
inventory
postgresql database
app.kubernetes.io/part-of : coolstore
app.kubernetes.io/instance : in
app.kubernetes.io/name : postgresql
app.kubernetes.io/component : database
app.kubernetes.io/managed-by : odo
app.kubernetes.io/version : 11
The node.js service with code
app.kubernetes.io/part-of : coolstore
app.kubernetes.io/instance : inventory
app.kubernetes.io/name : nodejs
app.kubernetes.io/component : frontend
app.kubernetes.io/managed-by : odo
app.kubernetes.io/version : 10
app.openshift.io/git-uri : git://git@github.com:gorkem/inventory-service.git
app.openshift.io/git-ref : master

TIP

It is a good practive to check if comination of `app.kubernetes.io/part-of`, `app.kubernetes.io/instance`, `app.kubernetes.io/name` labels lead to a meaningful identifier without repeating same parts. For instance `coolstore.catalog.mongodb`. A repeated part is usually an indicator of a label that can be avoided.