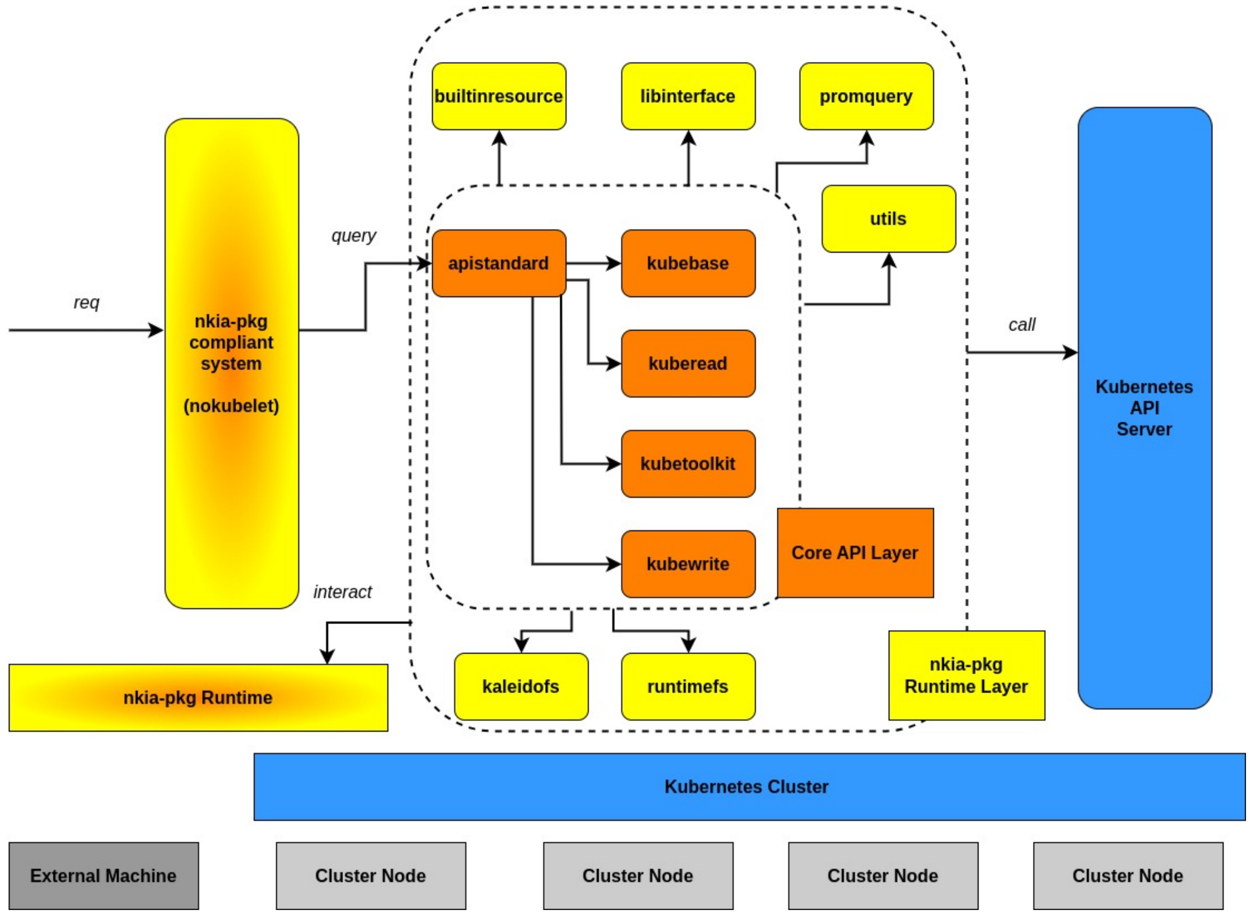
pkg



nkia-pkg overview

Overview

What this piece of code does is to offer the final execution layer of the functionalities targeted by the nokubectl(or its equivalent) user.
Using this piece of code, a user can read cluster node and application information, write desired lifecycle status to the cluster, which involves not only deployment but also updating, rolling-back, deletion and even qos management and auto-scaling, build and prepare the targets of deployment (or the applications) with user-supplied info into a Kubernetes-deployable apps, and configure and manage user-information without too much hassle on user side, all without actually interacting with kubectl or even kubeapi on user side

Functionalities

1 reading cluster node and application information (powered in part by Prometheus)

NKIA pkg provides the abstracted Kubernetes API that lets user to read
some of the most important information on Kubernetes cluster, including
node topology, node utilization, container status, container utilization
scheduled and unscheduled pods, etc. Adding to the array of useful information
made available by the native Kubernetes API, NKIA even supports flexible and
accurate time-series visualization base on Prometheus API

2 writing desired lifecycle status to the cluster, which involves not only deployment but also updating, rolling-back, deletion and even qos management and auto-scaling

NKIA pkg provides the core write functions regarding the managment of the
Kubernetes cluster and the deployment of project in a cloud-native way without
pedantically requiring user to implement all aspects of cloud-nativeness

All crucial Kunernetes native APIs that are used to manage container life-cycle
are abstracted and executable in the form of nokubectl command line
arguments (or, from web browser if it complies with the spec)

3 building and preparing the targets of deployment (or the applications) with user-supplied info into a Kubernetes-deployable apps (thanks to Kompose)

NKIA pkg supports out-of-the-box build and deployment pipepline based on the
docker-compose.yaml file presented at the root of the project git repository.
It automatically converts the docker-compose file into a deployable Kubernetes
resource file (thanks to Kompose) and caches it for further use.

4 configuring and managing user-information without too much hassle on user side

NKIA pkg can help user configure and manage his or her repository and image registry
credential across multiple clusters without having to write down where goes where
while not storing on the server-side(orch.io), thus preventing the single point of
failure in credential management, enhancing security.

How to use

Since NKIA pkg is, well, a set of common packages used across the whole NKIA project
using it directly involves importing it inside the target source base.

Here are the basic feature of the eact of the entry in NKIA pkg.

- pkg/apistandard
  Has the main entry point for all apis along the definition of all the api I/O data formats and structures of all available
  api commands.
- pkg/builtinresource
  Has the Kubernetes resource manifestations that are used in this project.
- pkg/kaleidofs
  Has all the functionalities to switch context between multiple Kubernetes cluster
  on the same node. ex) if a user wants to temporarily disable a specific cluster's
  connection from orch.io and manually manage it using native kubectl1
- pkg/kubebase
  Has the tools to set up a general Kubernetes management and deployment requirements.
- pkg/kuberead Has the tools to read from the cluster the desired information such as node, pod running status

at times with visualization available.

- pkg/kubetoolkit
  Has the tools to (primarily) build and prepare container images specified by user and turn it into a cluster deployable manifests.
- pkg/kubewrite
  Has the tools to write to clusters the information that is usually Kubernetes manifests under the hood.
- pkg/libinterface) Provides other functions with the interface to interact with the executables and scripts in lib
  directory.
- pkg/promquery
  Provides (primarily) pkg/kuberead and others with the interface to query to and retrieve standardized

  visualization data from Prometheus server.
- pkg/runtimefs
  Provides other functions with the interface to read and write from user-specific directory that holds information such as user Git and registry credential and cluster topology.
- pkg/utils
  Has the utilities that can be imported universally to handle granular tasks across the whole project.