nokubectl
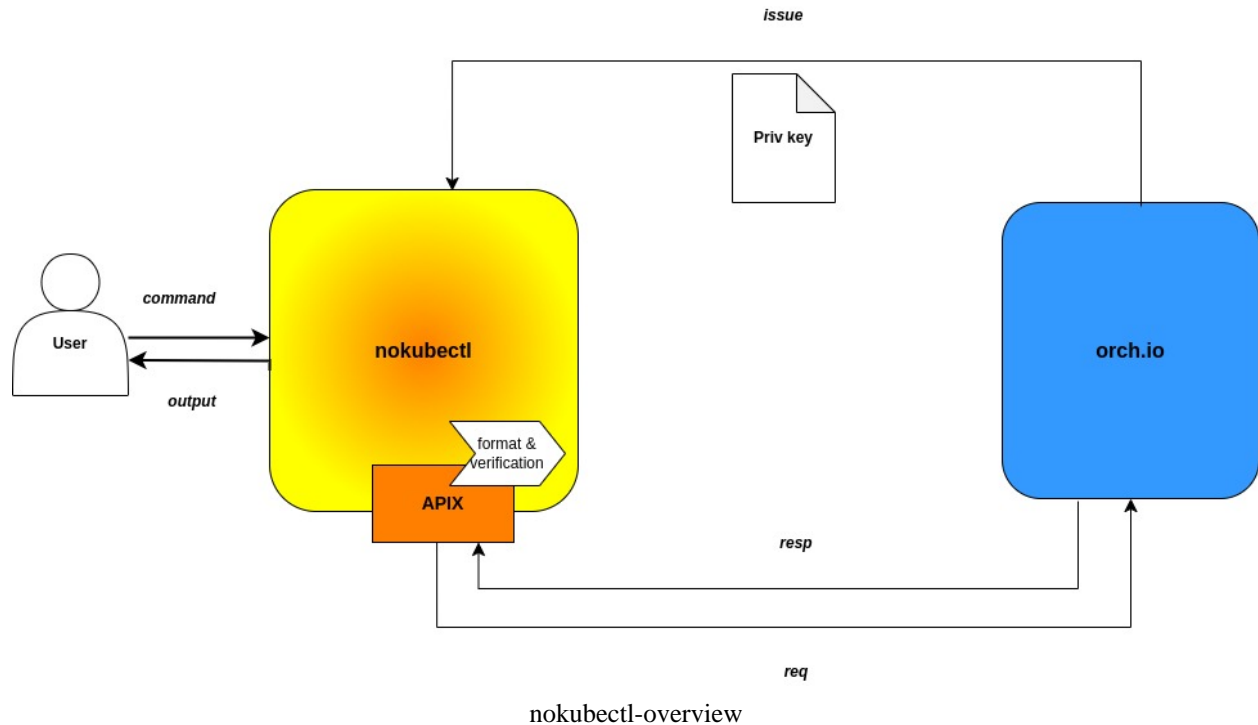
nokubectl-overview

Overview

What this piece of code does is to make a request to orch.io component of NKIA project,
receive the response, and output or cache the response in a user designated way for further use.
Using this piece of code, a user can have a default cmd interface that abstracts away
the complexity of multi-cluster environment and is easy to port to a web browser interface, opening
a room for more user and business friendly way of interacting with multi-cluster

Functionalities

1   Helps user make a request for managing container lifecycle across multiple Kubernetes clusters

It defines in its code the command line options available for a specific command a user might
want to execute on a particular cluster. All commands that are valid to send to orch.io and
their corresponding options are tranparently described in the code
(and will be made available online soon) and can be exported to other programming languages
other than Go (so that porting to web browser using Javascript or something might be easier)\

Within NKIA project, this system of API spec portability and
observability is called **APIX(APIeXtended)**

By harnessing the APIX system, a user of nokubectl doesn't have to worry about
the validity of the request

because APIX will handle the integrity check of the command and
even the construction of a query for the user.

> 2 Provides the guideline for embedding the multi-cluster abstration logic in other applications (ex, web browser interface)

While the first item describes the core functionality of nokubectl, the program itself is the explicit guideline for other programs that aim to replicate the multi-cluster command abstraction system that NKIA project supports

Just implements your own NodeJS, Python, C/C++, Rust, etc command line or web interface in accordance with the code represented in nokubectl source code

How to use

Precompiled binaries are available at here

Or, you can compile it by yourself using the following commands

```
git clone https://github.com/OKESTRO-AIDevOps/nkia.git
cd nkia
make release
# or
# `make build`
# if you just want to compile a binary without
# configuring essential environment
```

There are three requirements before actually using nokubectl

1. A running orch.io server
2. A private key file located at .npia/ directory which has been generated in advance from orch.io server
3. At least one Kubernete cluster or one server pool to become Kubernetes cluster, because NKIA project supports Kubernetes installation

To set up a running orch.io server, check out the documentation for that

Here