hack

Overview

What this piece of code does is to handle various Shell related tasks or help
functions that may not directly affect the execution of the core componentes
(nokubectl, nokubeadm, nokubelet, orch.io, pkg) but are crucial for resolving
dependencies and checking for overall integrity of the build output, etc.
Using this piece of code, a user or a developer can more easily benefit from
some of the most basic yet tedious tasks such as installing and configuring
test infrastructure including Kubernetes clsuter, Docker, CNI, Ingress, etc, while
also accessing the contents of raw scripts that are used when nokubeadm is running
to install and modify Kubernetes cluster, and many more.

Functionalities

1   Installing and configuring the test infrastructure for Kubernetes cluster

For example, scripts in hack/install directory contain various config files and
shell files that are used to set up a fully functioning, High Availability cluster.
Also, scripts in hack/kindcluster directory contain config files and shell files
that are used to create a Docker-based containerized Kubernetes node that can be
more easily and readily usable (but not fully functional)

2   Building (and updating) Library scripts when making the release files

Scripts in hack/libfactory contain the exact shells files that are used by the core
components via official release file lib.tar.gz from the git repo.
Regular Make build process calls the command internally to include all files inside
this directory.

3   Building release files for multiple operating systems on a single host

Files in hack/release are used to build multiple releases for multiple operating
system at the same time, harnessing the containerization technology powered by
Docker

4   etc

For example, files in hack/mdpdf are used to generate pdf files from these markdown
documents

How to use

By calling each script, a user can perform the task of his or her choosing.

For example, calling a specific script bears the following effect:

● hack/libgen.sh

Generates lib.tar.gz that is identical to the most recent version (if git HEAD

is latest) of lib.tar.gz on the git release repository

- hack/dep.sh

Installs requirements to build and release NKIA projects, for now, there are only Docker and Go for that

- hack/binupdate.sh

Checks and updates the locally installed core components