

# Kubeflow / ODH Disconnected

Václav Pavlín <[vasek@redhat.com](mailto:vasek@redhat.com)>

# Agenda

1. The WHY
2. Approaches
  - a. Preprocess manifests
  - b. Parameterize manifests
  - c. Provide a plugin
    - i. Explicit execution
    - ii. Automatic execution

# Disconnected deployment

- Disconnected or air-gapped clusters are common in
  - Government/Military
  - Finance
  - Edge
  - ...
- Need a way to be able to
  - get list of all images,
  - mirror all images into private registry,
  - transform manifests to use private registry

# Approach: Preprocessing manifests

1. Provide a tool which would iterate over all manifests in a repo
2. Each manifests with image is updated based on rules

The tool could also easily print all the images for mirroring.

This works OK-ish when `kfctl/kustomize` are invoked from commandline, but would add a manual step to the operator workflow.

# Approach: Parameterizing manifests

1. Parameterize all image references in all manifests (that is a lot of work)
2. User can use a parameter in KFDef to change the registry URL

A separate approach needs to be taken to get a list of all images.

This means a ton of work up front and even more over time for maintenance. It is also hard to verify all image references were covered since it is manual.

# Approach: Providing a plugin

1. Verify ODH/KF operator can be built in a way that supports plugins
2. Provide a plugin which transforms image references based on configuration
3. Make the plugin available in ODH/KF operator image

The plugin can have 2 functions - transforming manifests to use new registry URL and generating a list of images which are transformed - these can then be mirrored.

This is the best approach - it does not require additional steps for users and provides all the functionality needed in the native way

# Plugin: explicit execution

- The Kustomize plugins are executed based on being referenced in Kustomization.yaml
- This requires manifests maintainers to add the plugin reference to all Kustomization.yaml files

# Plugin: Automatic execution

- Provide a configuration option in KFDef which will make sure the plugin is enabled for all components automatically - i.e. will add the reference to the plugin to all Kustomization.yaml files

```
spec:
  global:
    transformers:
    - name: Registry Replace
  repoRef:
    name: manifests
    path: transformers/registry-replace.yaml
```



# Example Plugin Config

```
apiVersion: v1alpha1
kind: RegistryReplace
metadata:
  name: myTransformer
spec:
  fieldSpecs:
  - kind: Deployment
    group: apps
    version: v1
    path: spec/template/spec/containers[]/image
  - kind: ImageStream
    group: image.openshift.io
    version: v1
    path: spec/tags[]/from/name
  - kind: ImageStream
    group: image.openshift.io
    version: v1
    path: spec/tags[]/annotations/openshift.io\imported-from
  - kind: ConfigMap
    version: v1
    path: data/image
  - kind: ConfigMap
    version: v1
    path: data/someTextField
  registries:
  - src: quay.io
    dest: private-registry.example.com
  - src: docker.io
    dest: private-registry.example.com
```

# Example transformation

```
apiVersion: image.openshift.io/v1
kind: ImageStream
metadata:
  name: s2i-minimal-notebook
spec:
  tags:
  - annotations:
    openshift.io/imported-from: quay.io/thoth/minimal
  from:
    name: quay.io/thoth/minimal:v0.0.4
```

```
apiVersion: image.openshift.io/v1
kind: ImageStream
metadata:
  name: s2i-minimal-notebook
spec:
  tags:
  - annotations:
    openshift.io/imported-from: private.example.com/thoth/minimal
  from:
    name: private.example.com/thoth/minimal:v0.0.4
```

# What's next

- Turn the go plugin into exec plugin
  - go plugins will be [deprecated](#) anyway
- Improve the plugin to better handle the image replacement
  - `quay.io/a/b` vs. `a/b` vs. `b`
- Add global generators
  - `spec.global.generators`
- Figure out how to present the image list to users
  - Currently `/tmp/images.yaml`
- Propose to Kubeflow community
  - Community meeting? WGs?

# Code

- <https://github.com/vpavlin/kfctl/tree/feature/add-global-transformer>
- Plugin
  - <https://github.com/vpavlin/kfctl/blob/feature/add-global-transformer/kustomize-fns/v1alpha1/registryreplace/RegistryReplace.go>
- Manifests
  - [https://github.com/vpavlin/manifests/blob/feature/registry-replace-plugin/kfdef/kfctl\\_openshift.v1.2.0.yaml#L100](https://github.com/vpavlin/manifests/blob/feature/registry-replace-plugin/kfdef/kfctl_openshift.v1.2.0.yaml#L100)