

Promoting container images between registries with skopeo

August 3, 2018 | by Jorge Morales

OpenShift admins choose different architectures for their installations, but many use two discrete clusters to physically divide development and testing workloads from production deployments.

We recommend having some Continuous Integration (CI) process in nearly every development scenario, to orchestrate the lifecycle of applications from the initial commit all the way into production. Continuous Integration can imply many different tasks related to application development, testing, and releases. In this article, I won't directly address whether one should do Continuous Delivery or Continuous Deployment, as that depends on the maturity and complexity of the organization and the application.

No matter which technique you apply in your CI process, the end goal is for applications to move from development into production. In the world of containers running on orchestrated clusters, applications are packaged as container images and stored in registries. So promoting an app from development to production often means copying a container image from one registry to another. The [OCI Image Spec](#) gives us a standard format for container images, making such OCI container images portable between container registries and runtimes.

At Red Hat, we've been working for some time on tools for handling OCI images. The OCI specification means that an image can be created, stored, or executed with any tool that adheres to the spec. Whether you want to build an image with [Buildah](#), or run it on a Kubernetes/OpenShift cluster using [CRI-O](#), or keep it old school and do both jobs with Docker, the image spec means the choice is up to you. OCI-standard container images can be shared among all of those tools.

When it comes to inspecting and transporting images, a tool called [skopeo](#) proves very useful. Skopeo inspects container images in any of the places where an OCI image can be stored. It can also copy container images from one location to another. If you want to copy an image from your laptop's local docker storage to the local CRI-O container store, it's as easy as:

```
skopeo copy docker-daemon:myregistry/myimage:1.0.0 container-storage:myregistry/myimage:1.0.0
```

Skopeo is also useful for [copying images between two remote docker registries](#), such as the registries of two different OpenShift clusters.

```
skopeo copy docker://clusterA:5000/myregistry/myimage:1.0.0 docker://clusterA:5000:myregistry/myimage:1.0.0
```

What's so revolutionary about copying container images around? Well, the trick is that skopeo is a standalone tool that doesn't require a docker --or any other-- daemon, so it can easily be used in Continuous Integration pipelines.

Here's an example of a CI pipeline:

```

def namespace, appReleaseTag, webReleaseTag, prodCluster, prodProject, prodToken
pipeline {
  agent {
    label 'skopeo'
  }
  stages {
    stage('Choose Release Version') {
      steps {
        script {
          openshift.withCluster() {
            // Login to the production cluster
            namespace = openshift.project()
            prodCluster = env.PROD_MASTER.replace("https://", "insecure://")
            withCredentials([usernamePassword(credentialsId: "${namespace}-prod-credentials",
            usernameVariable: "PROD_USER", passwordVariable: "PROD_TOKEN")]) {
              prodToken = env.PROD_TOKEN
            }

            // Get list of tags in the ImageStream to show the release-manager
            def appTags = openshift.selector("istag").objects().collect { it.metadata.name }.findAll
            { it.startsWith 'app:' } .collect { it.replaceAll(/app:(.*)/, "\$1") }.sort()
            timeout(5) {
              def inputs = input(
                ok: "Deploy",
                message: "Enter release version to promote to PROD",
                parameters: [
                  string(defaultValue: "prod", description: 'Name of the PROD project to create', name:
                    'PROD Project Name'),
                  choice(choices: appTags.join('\n'), description: '', name: 'Application Release
                    Version'),
                ]
              )
              appReleaseTag = inputs['Application Release Version']
              prodProject = inputs['PROD Project Name']
            }
          }
        }
      }
    }
    stage('Create PROD') {
      steps {
        script {
          openshift.withCluster(prodCluster, prodToken) {
            openshift.newProject(prodProject, "--display-name='CoolStore PROD'")
          }
        }
      }
    }
    stage('Promote Images to PROD') {
      steps {
        script {
          openshift.withCluster() {
            def srcApplicationRef = openshift.selector("istag",

```

```

"app:${appReleaseTag}").object().image.dockerImageReference
def destApplicationRef = "${env.PROD_REGISTRY}/${prodProject}/app:${appReleaseTag}"
def srcToken = readFile "/run/secrets/kubernetes.io/serviceaccount/token"
sh "skopeo copy docker://${srcApplicationRef} docker://${destApplicationRef} --src-creds
openshift:${srcToken} --dest-creds openshift:${prodToken}"
}
}
}
}
stage('Deploy to PROD') {
  steps {
    script {
      openshift.withCluster(prodCluster, prodToken) {
        openshift.withProject(prodProject) {
          def template = 'https://raw.githubusercontent.com/openshift-labs/myapp/myapp-
template.yaml'
          openshift.apply(
            openshift.process("-f", template, "-p", "APPLICATION_IMAGE_VERSION=${appReleaseTag}", "-
p", "IMAGE_NAMESPACE=")
          )
        }
      }
    }
  }
}
}
}
}
}
}
}
}
}
}

```

In our [OpenShift workshops](#), we use this pipeline to illustrate Continuous Delivery onto the platform. I could go on about Continuous Delivery best practices, but today's post focuses on skopeo, the key to how this pipeline copies container images between two clusters' registries.

With skopeo, it's also easy to provide authentication credentials for secured registries. There are different mechanisms to supply your credentials, and all of them are easy to embed in a pipeline or to use directly from the command line.

If you need to move container images between public registries or to promote images from a dev registry into prod, try out skopeo. Skopeo is a stable tool with a track record of extensive use at Red Hat over the last year, but if you run into problems, you can report them directly to the developers at the [project's GitHub repository](#).

In a world where container images are compliant with the OCI Image Specification, skopeo is a daemonless tool that helps you copy images between different storage locations. Whether this storage is local to a node, like docker-storage and CRI-O storage, or it's remote, such as in a copy from one Docker registry to another, skopeo is the tool for the job.

Brand new to OpenShift? Check out the interactive tutorials complete with a live, hosted OpenShift environment at [learn.openshift.com](#). Experienced app dev on the platform? Keep up to date and informed with the [OpenShift documentation](#), or expand your OpenShift environment to Red Hat-hosted [OpenShift Online](#) or on-premises [OpenShift Container Platform](#).