

lilpenguin - Justin's rants & raves:"JUST want to pull & run an image from Docker Hub! Why does OpenShift make this so hard!?"

Posted by [Justin Pittman](#) Jan 25, 2019

- [Problem Statement](#) on page 1
- [KISS Solution](#) on page 1
- [Non-Starters](#) on page 3
 - [Our Customer Portal](#) on page 3
 - [Our Container Catalog](#) on page 3
 - [Our OpenShift Blog / Commons](#) on page 3
 - [Public Answers Fall Flat](#) on page 4
 - [S2I Isn't For EVERYone!](#) on page 4
 - [No Magic Here](#) on page 5
- [But Why?! on page 5](#)
- [Conclusions](#) on page 5

Problem Statement

Yesterday, a technology partner (whose well known in the Java world) admitted in email a frustration that other partners & customers have hinted at when I've asked their engineers to deploy their own, prebuilt image from Docker Hub into OpenShift:

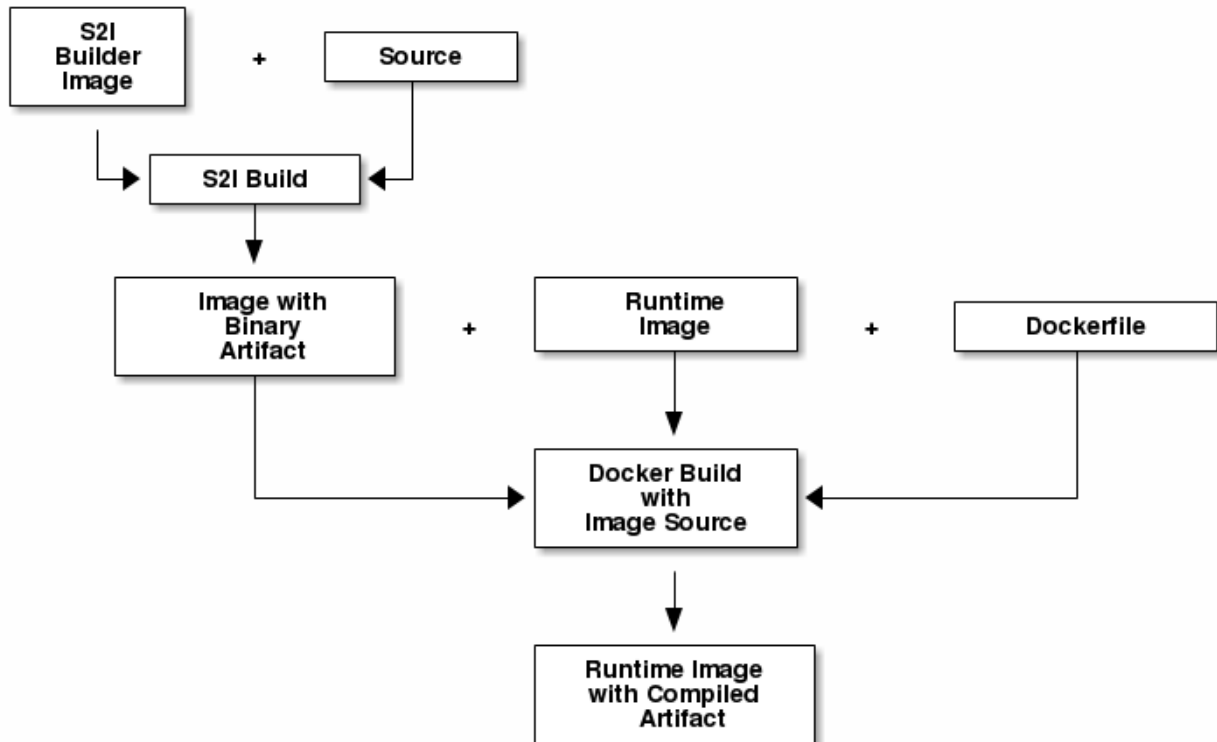
I find openshift [...] just add complexity to deployments.

Just a few days later, another customer (small shop but smart folks) reached out after they got misled by Googling their way through how to deploy their pre-built docker image into OpenShift. 🙄

KISS Solution

Let's get to the point and elaborate later. Here's the solution in a diagram (posted by [Jorge Morales's Blog](#))

lilpenguin - Justin's rants & raves: "JUST want to pull & run an image from Docker Hub! Why does OpenShift make this so hard!?"



And here's how OpenShift users, who're coming from Docker tooling or comfortable with docker CLI, will get OpenShift CLI (oc client) commands doing basically the same thing as docker login, docker pull, docker run, etc. when interacting with Docker Hub:

To pull & run/deploy:

```
oc new-app sonatype/nexus-3
```

or to pull only:

```
oc import-image sonatype/nexus-3 --confirm
```

To specify image location (in container registry) for pull & run, with container parameters injected at runtime:

```
oc new-app --docker-image=registry.access.redhat.com/redhat-sso-7/sso72-openshift \
-e SSO_ADMIN_USERNAME=ssoadmin \ -e SSO_ADMIN_PASSWORD=redhat \
-e SSO_REALM=Training-lab \ -e SSO_SERVICE_USERNAME=student \ -e SSO_SERVICE_PASSWORD=redhat \
--name=sso-img
```

or to specify Docker Hub (or any remote registry) to just pull from:

```
oc import-image sonatype/nexus-3 --from=docker.io/sonatype/nexus-repository-manager --confirm
```

lilpenguin - Justin's rants & raves: "JUST want to pull & run an image from Docker Hub! Why does OpenShift make this so hard!?"

So there's a KISS for folks looking.

Non-Starters

Our Customer Portal

"Don't Google!", you say? OK, when I searched our Customer Portal for answers, like a customer/partner would do if they used our website, by searching for phrases like:

"external registry"

"pull image from external"

"pulling from Docker Hub"

Our answers are long-winded KB articles that define terms, like *Image Streams*, and explain processes, like image *Build Strategies*, or the massive Dev Guide; instead of KISS (Keeping It Simple, Stupid). KB examples:

- [Builds and Image Streams - Core Concepts | Architecture | OpenShift Container Platform 3.11](#)
- [OpenShift Container Platform 3.11 Developer Guide - Red Hat Customer Portal](#)
- [Red Hat Guide to Linux Containers - Red Hat Customer Portal](#)

Our Container Catalog

We do actually give a succinct step via OpenShift CLI (oc client) commands, so a KISS, on the RH Container Catalog but that means a customer/partner would need to be looking for images *that we host, instead of how to interact with Docker Hub*. For example, KISS method here and its related docker CLI counterparts could be found, for the very persistent customer/partner, on the Container Catalog under any image: [Container Catalog - Red Hat Customer Portal](#)

Our OpenShift Blog / Commons

Our very own Veer does actually demo how to deploy an image from Docker Hub -- he did it wayyyyyy back in 2015! When OpenShift v3 was in Beta!! [OpenShift 3 Beta Demo Part 2: Deploying MySQL Using a Docker Image - YouTube](#). We don't even use the CLI (osc) anymore that is in that demo, but I've found no updated demo from us since. Hmm.

RHers may assume our public/private Workshops and OpenShift UI (Image Stream UI, Template UI) are "better" ways of enabling customers/partners but this technical marketing/enablement doesn't show how to accomplish tasks docker tooling tasks, so 'docker pull' or 'docker run', in OpenShift. Neither the OpenShift

lilpenguin - Justin's rants & raves: "JUST want to pull & run an image from Docker Hub! Why does OpenShift make this so hard!?"

Java Workshop nor the DevOps Workshop showcase the above OC client commands, even though the CLI is occasionally used in these workshops.

- [Workshopper](#)
- [DevOps Workshop with OpenShift](#)

The OpenShift blog authors gave succinct, excellent steps after searching that website, and these authors got to a similar KISS point, but those blogs were from 2016! This is 2019!! Who can *actually* find these today?!

- [Getting any Docker image running in your own OpenShift cluster – Red Hat OpenShift Blog](#)
- [Deploying Images from Docker Hub – Red Hat OpenShift Blog](#)
- [Remotely Push and Pull Container Images to OpenShift – Red Hat OpenShift Blog](#)
- [Binary Deployments with OpenShift 3 – Red Hat OpenShift Blog](#)

So customers/partners who find or are shown these blogs and the demo begin to wonder whether OpenShift is up-to-date, while internal engineering has moved so far ahead, with Operators, Service Meshes, CRI-O, etc., that the customer gets left behind.

Public Answers Fall Flat

The public does try to reverse engineer how we accomplish simple tasks, like pulling an image from Docker Hub. Here's a blog from 2017 that I found while Googling that uses a *whopping* 7-8 commands when one `oc` command would suffice: <https://medium.com/@adilsonbna/importing-an-external-docker-image-into-red-hat-openshift-repository-c25894cd3199>. And at least one StackOverflow reply said tasks like this aren't even possible, which is obviously *wrong* but our own Graham Dumpleton's reply wasn't up-voted (though in fairness this StackOverflow question was about importing multiple images at once but the OP didn't make that clear): [How to pull docker images from public registry and push it to private openshift? - Stack Overflow](#). So public answers fall flat and are probably creating bad optics for us.

S2I Isn't For EVERYone!

More than one customer/partner has told me they can't (yet) change their tooling to accommodate building from source, yet you'll find we talk about source-to-image (S2I) ALL THE TIME! We mention other build methods as 2nd class citizens. Take this wording as an example: "In this process we won't use S2I but instead use a docker file ..." ([Binary Deployments with OpenShift 3 – Red Hat OpenShift Blog](#)) So S2I is a 1st class citizen for us. Building mindshare by repetition is fine, but S2I builders aren't for everyone. A customer in PoC/Eval of OpenShift that has pre-built container images to deploy sees no immediate value in all our messaging around S2I. And our constant messaging on S2I can bring bad optics.

Example: one customer exemplified how skewed our messaging has become by asking, "if we don't do S2I, Red Hat won't support us??" 🤖

(Did I type "S2I" enough in the previous paragraph? That was intentional.)

lilpenguin - Justin's rants & raves: "JUST want to pull & run an image from Docker Hub! Why does OpenShift make this so hard!?"

No Magic Here

The studious engineer will note that one of OpenShift CLI commands above is only given a relative parameter to find the image (i.e. "sonatype/nexus-repository-manager" instead of the absolute URL of "docker.io/sonatype/nexus-repository-manager"). There's no magic here, but we don't necessarily need to burden customers with this detail until they ask to reconfigure it. The answer is still simple -- OpenShift configuration of Docker Hub as a "trusted", remote repositories, aka the configuration options below -- but we can mention this detail as a footnote for transparency's sake:

The docker client config, shipped with OpenShift

```
cat /var/lib/origin/.docker/config.json
```

The standard docker daemon config, shipped with OpenShift

```
cat /etc/sysconfig/docker
```

The RHEL preferred way to configure the standard docker daemon

```
cat /etc/docker/daemon.json
```

But Why?!

Simply said: kubernetes does not include tooling to build and run containers, and so each "flavor" of kubernetes conformant platforms implement the workflow differently. Kubernetes clusters do indeed (almost) always run containers / Docker, but these are not part of Kubernetes by definition.

Conclusions

Even though it is 2019, when we're championing Operators and CRI-O and we've expanded the Service Catalog with pre-built templates, these technologies are not always where customers/partners are starting from as a baseline, especially for the CLI savvy. IMO, *this is another example of where we're moving at Ludicrous Speed, evangelizing about microservices and serverless, and assumed everyone already got the OpenShift basics.* For customers/partners coming from docker tooling (which is, from where I sit, a bigger market for us than folks coming from greenfield Cloud-Native or other tooling like Cloudfoundary/Pivotal), we've made it so difficult for them to find answers to the simplest of problems that they get frustrated and blame the product for being overly complex. We say the product is flexible and customizable; they say complex or bloated.

lilpenguin - Justin's rants & raves:"JUST want to pull & run an image from Docker Hub! Why does OpenShift make this so hard!?"

Like I said earlier, customer / partners begin to wonder whether OpenShift is right for them when the simplest tasks aren't clearly explained by us, or we explained the processes ages ago, while internal engineering and marketing has moved so far ahead, with Operators, Service Meshes, CRI-O, etc., that the customer gets left behind. Luckily, I caught this one but how many more just abandon OpenShift? Or do we not want these opportunities?

124 Views Tags: [oc](#), [openshift3.11](#), [openshift ecosystem](#), [docker pull](#)

There are no comments on this post