



Multi-Arch Layered Image Build System

PRESENTED BY:

Adam Miller

Fedora Engineering, Red Hat

Today's Topics

- Define “containers” in the context of Linux systems
 - Brief History/Background
 - Container Implementations in Linux
- Base Image vs Layered Image
- Why Fedora Containers?
- Why Multi-Arch Containers?
- Fun history lesson
- What is OpenShift?
- Define Release Engineering
- How does this all work?
 - How does it work today?
 - How will it work with multi-arch?

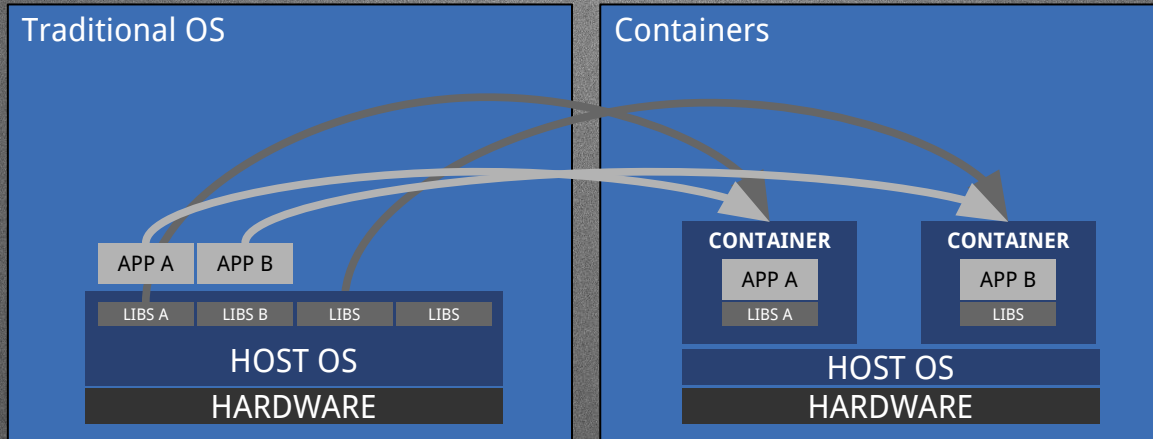




Containers

What are containers?

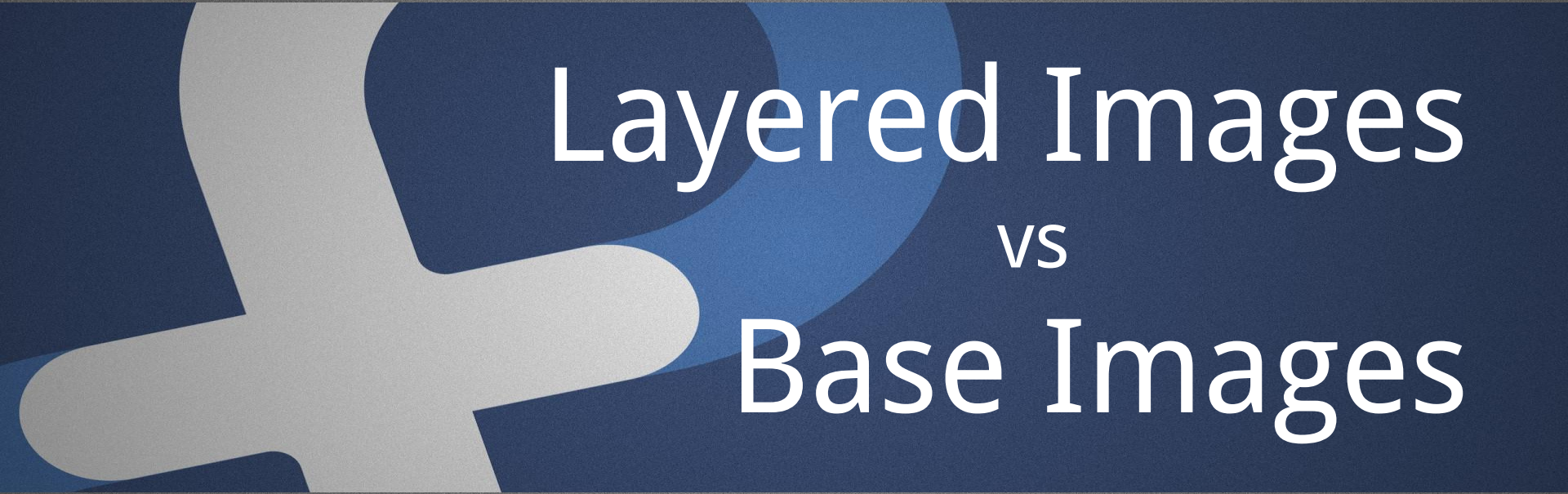
- Operating-system-level Virtualization
 - We (the greater Linux community) like to call them “containers”
- OK, so what is Operating-system-level Virtualization?
 - The multitenant isolation of multiple user space instances or namespaces.



Containers are not new

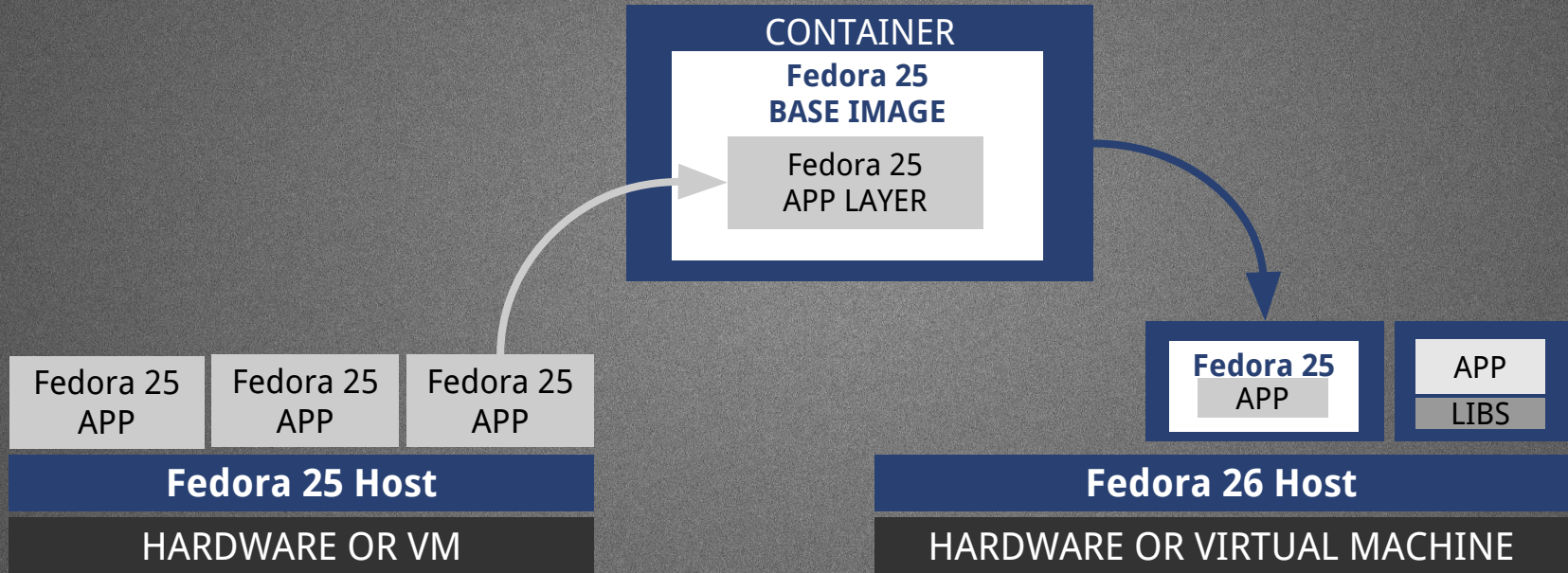
- The concept of containers is not new
 - chroot was the original “container”, introduced in 1982
 - Unsophisticated in many ways, lacking the following:
 - COW
 - Quotas
 - I/O rate limiting
 - cpu/memory constraint
 - Network Isolation
 - Brief (not exhaustive) history of sophisticated UNIX-like container technology:
 - 2000 - FreeBSD jails
 - 2001 - Linux Vserver
 - 2004 - Solaris Zones
 - 2005 - OpenVZ
 - 2008 - LXC (This is where things start to get interesting)
 - 2011 - Systemd nspawn
 - 2013 - dotCloud releases Docker (later renames itself to Docker Inc)
 - 2015 - runC is released under the purview of Open Container Initiatives
 - 2016 - containerd - runC orchestration daemon





Layered Images vs Base Images

Base vs Layered Images



An abstract graphic on the left side of the slide, featuring a light blue stylized shape that resembles a lowercase 'f' or a similar character, set against a dark blue background. The shape is composed of several rounded, overlapping forms.

Why?

Why.... ?

Why Fedora Containers?

- Delivering Fedora Content faster to users
- Automatically generating release artifacts with security updates
- Lowering the barrier of entry for contributors
 - Note: There are some obstacles here but plans in place to get past them

Why Multi-Arch?

- Fedora isn't just a x86_64 distro
- Internet of Things (IoT)
- "The Magical ARM Revolution"
- Other architectures matter!
- We don't know what's next, best not to box ourselves in.



The image features a dark blue horizontal band across the center. On the left side of this band, there is a large, light grey abstract shape that resembles a stylized human figure or a hand. To the right of this shape is a medium-sized, light blue circle. The text "History Lesson" is written in a white, sans-serif font, positioned to the right of the light blue circle and centered vertically within the dark blue band.

History Lesson

How FLIBS happened.

- Matt Miller, Fedora's Fearless Leader (Fedora Project Leader)
 - "There's this open source layered image build system I heard about, we should deploy one!" (Paraphrasing)
- Initial discussions estimated about four weeks of work to deploy a new service in the Fedora Infra and tie it into various build, test, and messaging services.
 - There was an incorrect assumption that it was a finished product



OSBS History

- Phase 1
 - Single-Node builder
 - Finished in a few months
 - Image Format v2, Registry v2, Manifest v2 - this broke the original implementation
- Phase 2
 - Scale-out deployment
 - Fully compat with Image Format v2, Manifest v2, Registry v2
 - Automated tests can be tied to the output of OSBS
 - RelEng is able to then "promote" images to a "production" or "stable" registry/tag/repository
- Phase 3 (Happening Now)
 - Image Registry Scale-out - Done
 - Search/Advertise image registry - In flight
 - CVE/Security metadata for updates - Planning
- Phase 4
 - Orchestrator/Worker Architecture
 - Multi-Arch



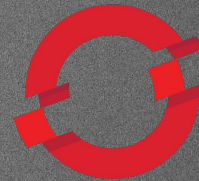
The image features a dark blue horizontal band across the center. On the left side of this band, there is a stylized logo consisting of a light grey 'K' shape and a blue circular element. To the right of the logo, the word 'OpenShift' is written in a white, sans-serif font.

OpenShift

OpenShift

- OpenShift
 - Container Platform built on top of Kubernetes
 - Advanced Features
 - Build Pipelines
 - Image Streams
 - Application Lifecycle Management
 - CI/CD Integrations
 - Binary Deployment
 - Triggers (Event, Change, Image, Web, etc)
 - REST API, Command line interface, IDE Integrations
 - Web UI and Admin dashboard

OPENSHIFT
origin

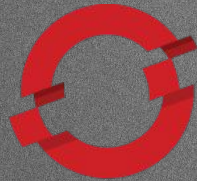


OPENSHIFT



OpenShift

OPENSHIFT
origin



OPENSHIFT[™]
by Red Hat[®]

CONTAINER

CONTAINER

CONTAINER

CONTAINER

CONTAINER

SELF-SERVICE

SERVICE CATALOG
(LANGUAGE RUNTIMES, MIDDLEWARE, DATABASES, ...)

BUILD AUTOMATION

DEPLOYMENT AUTOMATION

APPLICATION LIFECYCLE MANAGEMENT
(CI / CD)

CONTAINER ORCHESTRATION & CLUSTER MANAGEMENT
(KUBERNETES)

NETWORKING

STORAGE

REGISTRY

LOGS &
METRICS

SECURITY

INFRASTRUCTURE AUTOMATION & COCKPIT

CONTAINER RUNTIME & PACKAGING
(DOCKER)

ATOMIC HOST

Fedora / CentOS / Red Hat Enterprise Linux



Physical



Virtual



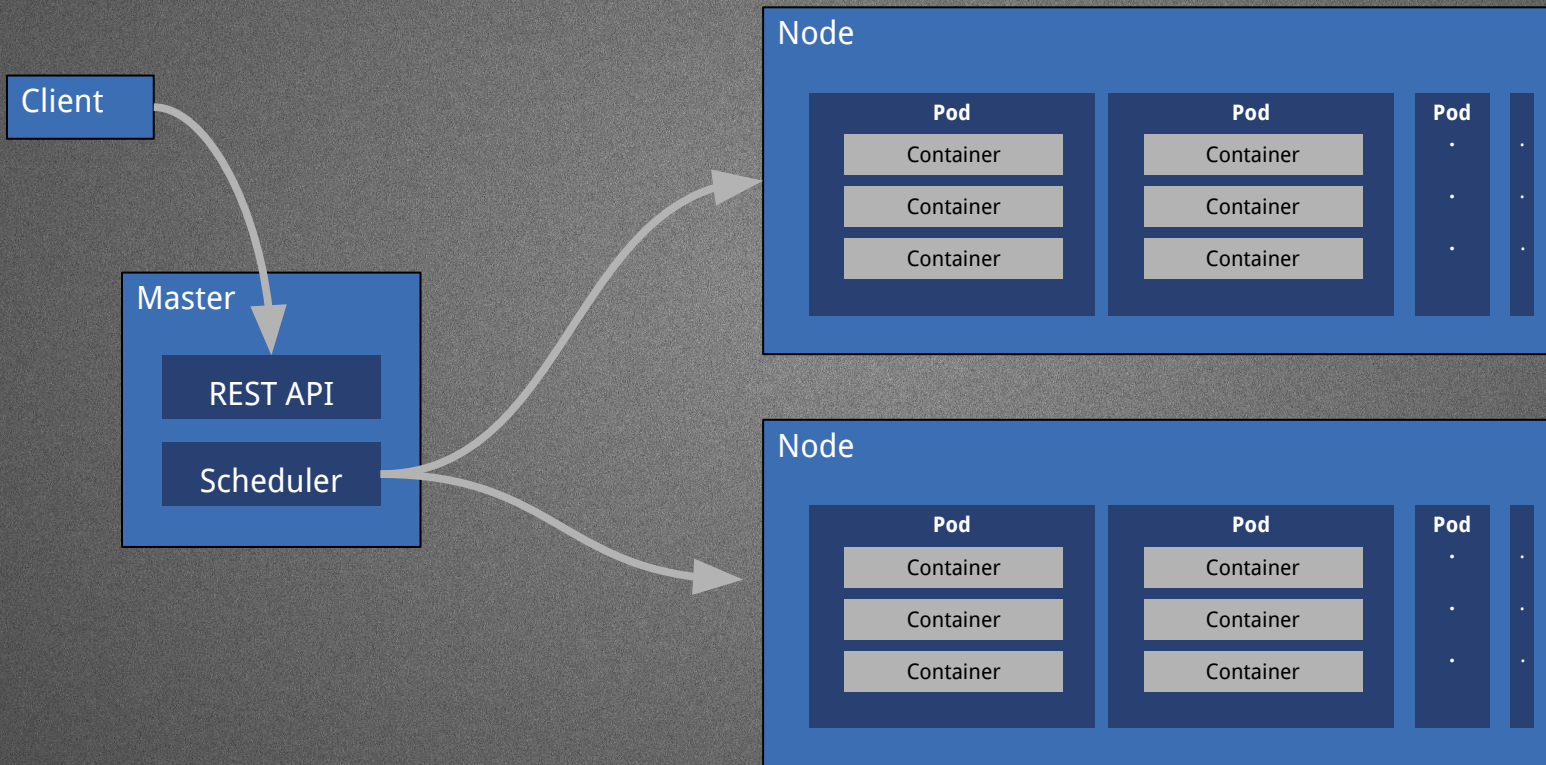
Private



Public



OpenShift/Kubernetes Overview



An abstract graphic on the left side of the slide, featuring a dark blue background with a light grey, stylized shape that resembles a person or a figure. The shape is composed of several rounded, overlapping forms. To the right of this graphic, the text "Release Engineering" is displayed in a clean, white, sans-serif font, stacked in two lines.

Release Engineering

Release Engineering

- What is Release Engineering?
 - Making a software production pipeline that is Reproducible, Auditable, Definable, and Deliverable
 - It should also be able to be automated
- Definition (or the closest there really is)

“Release engineering is the difference between manufacturing software in small teams or startups and manufacturing software in an industrial way that is repeatable, gives predictable results, and scales well. These industrial style practices not only contribute to the growth of a company but also are key factors in enabling growth.”

- Boris Debic of Google Inc



A stylized graphic on the left side of the banner. It features a light gray hand with fingers spread, holding a dark blue globe. The globe is partially obscured by the text. The background of the banner is a dark blue gradient.

Layered Image Build Service

OSBS

- OpenShift Build Service
 - Takes advantage of OpenShift's built in Build primitive with a "Custom Strategy" and BuildConfig
 - This defines what can be the inputs to a build
 - Relies on OpenShift for scheduling of build tasks throughout the cluster
 - Presents this defined component to developers/builders as CLI and Python API
 - osbs enforces that the inputs come from auditable sources.
 - Git repo for source Dockerfile, git commits and builds centrally logged
 - BuildRoot - limited docker runtime
 - Firewall constrained docker bridge interface
 - Unprivileged container runtime with SELinux Enforcing
 - Inputs are sanitized before reaching to build phase
 - Unknown or unvetted sources are disallowed by the system
 - Uses OpenShift ImageStreams as input sources to BuildRoot
 - Planned utilization of OpenShift Triggers to spawn rebuild actions based on parent image changes
 - Factory 2.0 will also launch new builds for when RPM content changes

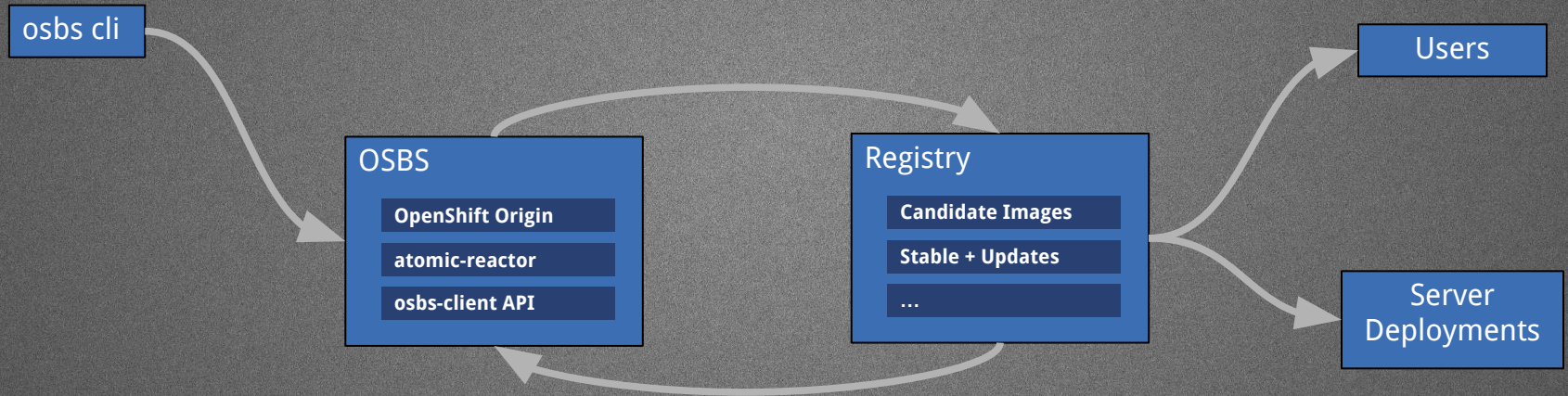


OSBS - Continued

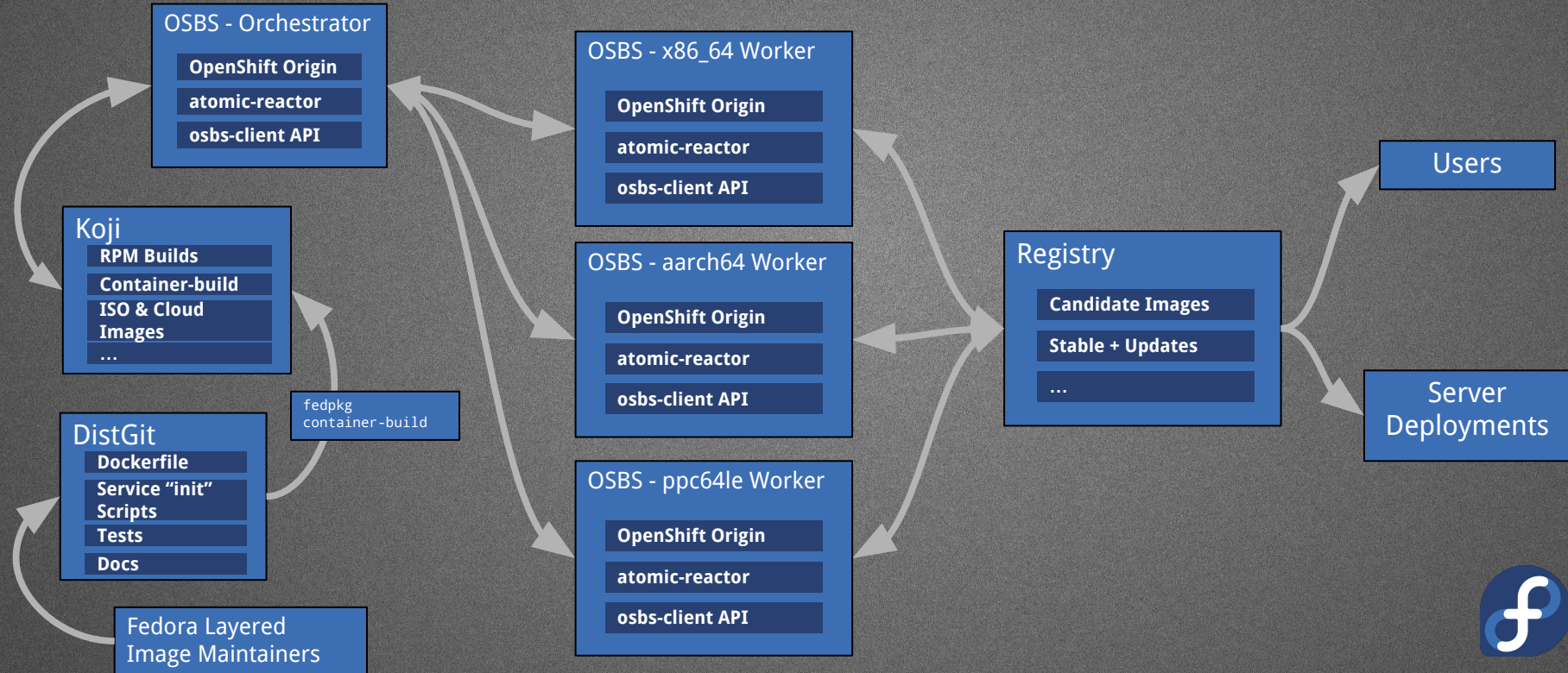
- atomic-reactor
 - Single-pass Docker build tool used inside constrained buildroot in OSBS
 - Automates tasks via plugins, such as:
 - pushing images to a registry when successfully built
 - injecting yum/dnf repositories inside Dockerfile (change source of your packages for input sanitization/gating)
 - change base image (FROM) in your Dockerfile to
 - match that of the registry available inside the isolated buildroot, run simple
 - tests after image is built
- Gating of updates
 - Automated tests can be tied to the output of OSBS
 - RelEng is able to then "promote" images to a "production" or "stable" registry/tag/repository



Build System



Multi-Arch Build System



Fedora's Implementation

- DistGit (“Distro Git”)
 - Each Branch = Fedora Release
 - master branch is Devel (codename “Rawhide”)
- fedpkg
 - Fedora Package Maintainer helper tool
 - Manages distgit branches
 - Initiate builds (local and remote, mock integration)
 - Much more ...
- Koji
 - Fedora’s authoritative build system
 - Everything for Fedora is built here or it’s build is integrated here
 - Live USB images, DVD ISOs, IaaS Cloud Images, RPMs, Docker
 - This defines what can be the inputs to a build
- Registry
 - Upload/download destination, point of distribution
- Orchestrator
 - OpenShift Cluster that orchestrates the builds across the arch-specific
 - Koji point of contact
- fedpkg
 - Fedora Package Maintainer helper tool
 - Manages distgit branches
 - Initiate builds (local and remote, mock integration)
 - Much more ...
- Koji-containerbuild
 - Plugin to orchestrate builds between Koji and OSBS



Lessons Learned

- The OSBS Upstream Team is fantastic
- The OpenShift Team is also fantastic
- OpenShift is really powerful
- The container technology space moves **fast**
- Nothing is set in stone
 - Don't expect APIs to remain relevant
 - Don't expect backwards compatibility
- People are starting to care about architectures other than x86_64
 - (They have for a while, but now it's gaining traction)



An abstract graphic on the left side of the slide, featuring a large white shape that resembles a stylized letter 'X' or a similar symbol, set against a background of various shades of blue. The shapes are smooth and rounded, creating a modern, geometric aesthetic.

Questions?

CONTACT:
maxamillion@fedoraproject.org
[@TheMaxamillion](https://twitter.com/TheMaxamillion)

References

- https://en.wikipedia.org/wiki/Operating-system-level_virtualization
- <https://coreos.com/blog/rocket>
- <https://coreos.com/blog/appc-gains-new-support>
- <https://www.docker.com>
- <https://github.com/docker/distribution>
- <http://www.redhat.com/en/insights/containers>
- <http://www.projectatomic.io>
- <http://www.openshift.org>
- <https://www.openshift.com>
- <http://www.redhat.com/en/about/blog/red-hat-and-google-collaborate-kubernetes-manage-docker-containers-scale>
- <http://rhelblog.redhat.com/2014/04/15/rhel-7-rc-and-atomic-host>
- <http://opencontainers.org>
- <http://runc.io>
- <http://queue.acm.org/detail.cfm?id=2884038>
- <http://containerd.tools>
- https://drive.google.com/file/d/0B_Jl94nModqdSFVseUotQVB1Rnc/view?usp=sharing
- <http://valleyproofs.debic.net/2009/03/behind-scenes-production-pushes.html>
- <https://github.com/release-engineering/koji-containerbuild>
- <https://github.com/projectatomic/atomic-reactor>
- <https://pagure.io/koji>
- <https://fedorahosted.org/koji/wiki>
- https://bugzilla.redhat.com/show_bug.cgi?id=1243736
- https://fedoraproject.org/wiki/Changes/Layered_Docker_Image_Build_Service
- <https://osbs.readthedocs.io/en/latest/>
-

