



## Tom Larrow

DevOps Automation Engineer at Key Bank

---

# Private Cloud Deployment Automation with XebiaLabs and OpenShift

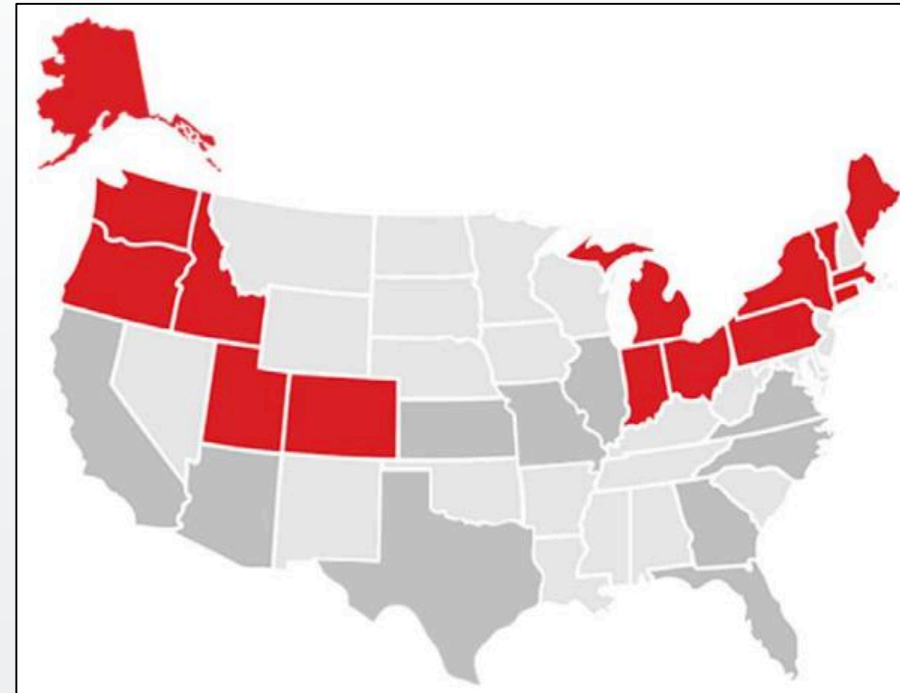


# About KeyBank

- One of the **Top 25** largest banks in US
- Headquartered in **Cleveland OH**
- \$136 Billion in Assets
- \$5.8 Billion in Revenue
- 20,000+ Employees
- 2 Datacenters in US
- 8 consecutive **Outstanding ratings for lending** under the Community Reinvestment Act
- 4 times named a **Best Place to Work for LGBT Equality** by the HRC

## Corporate Bank Business Units

- Real Estate Capital
- Treasury Management
- Derivatives/Foreign Exchange
- Investment Banking
- Equipment Finance
- Institutional Asset Services
- Broker Dealer
- Pacific Crest Research





# Impetus for Change

---

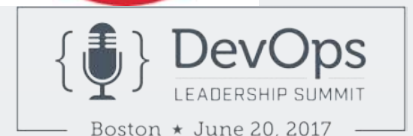
- Major outage revealed complexity of systems
  - One online banking login required 190 Network Hops across 2 datacenters
- Lack of Standard Configurations and Automation allowed only quarterly releases
  - All hands on deck 4 times a year
- Acquisition of First Niagara
  - Needed to update online banking sooner than anticipated



# Updating our Platform

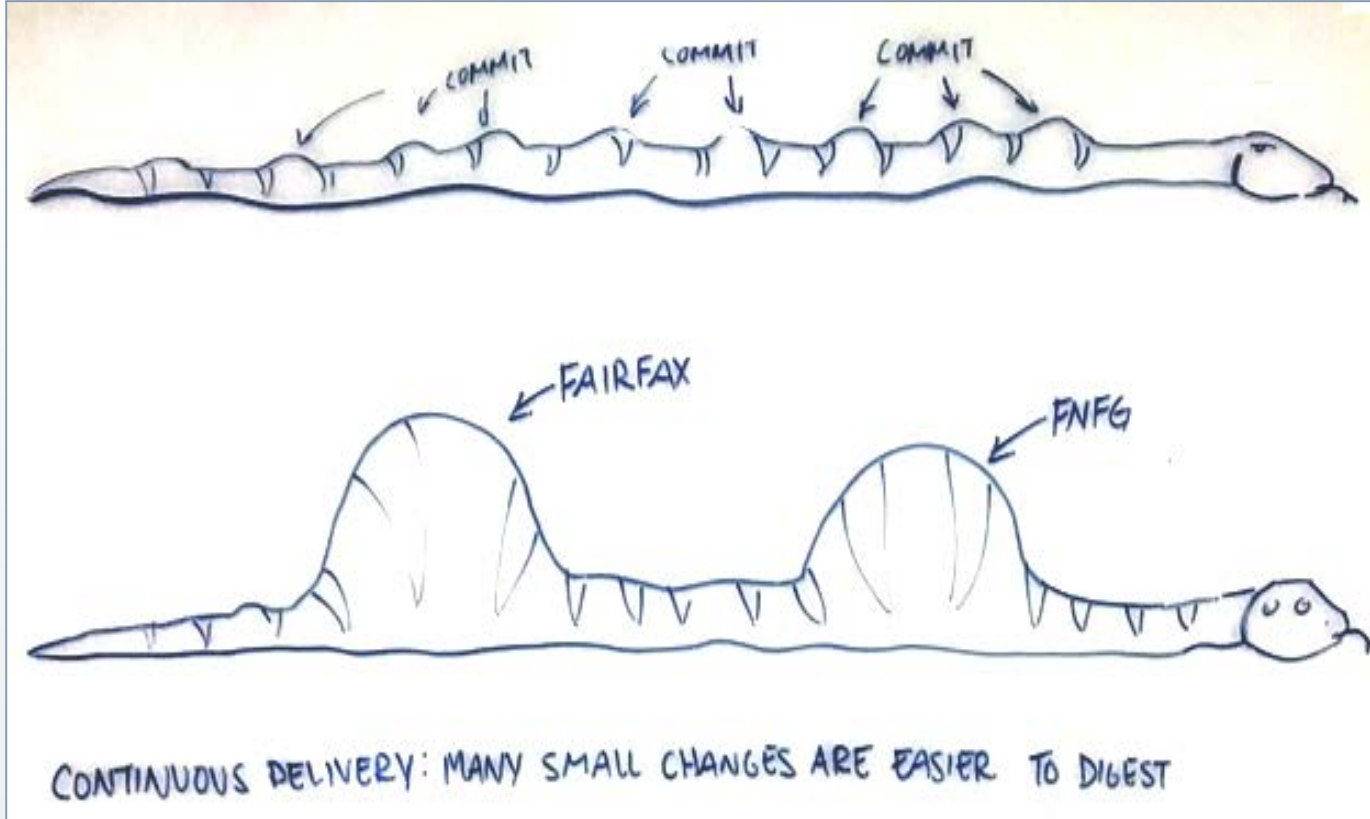
---

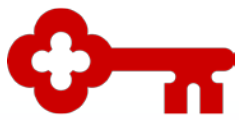
- Moved to Docker containers for immutable infrastructure
  - Dramatically decreased time to provision a server
  - Run inside the same container that was tested
- Kubernetes for container orchestration
  - Insure containers always up, in optimum amount to handle load
  - Technology that powers Google
- Red Hat OpenShift
  - Red Hat supported version of Kubernetes
  - Self Serve PAAS abilities



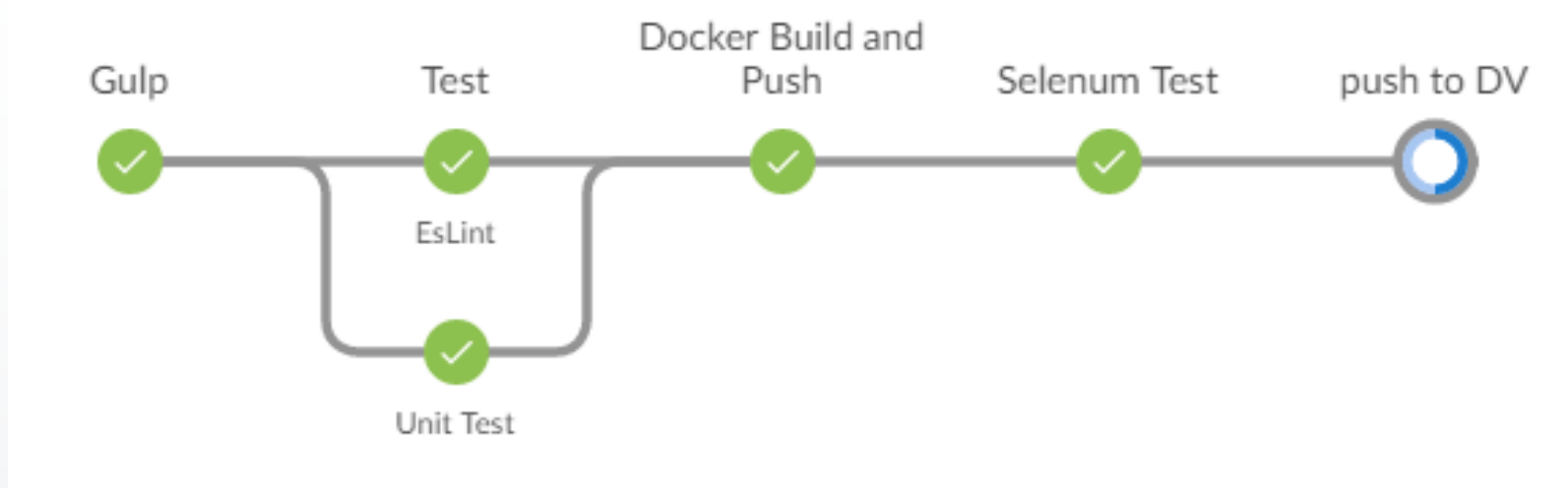


# Updating our Processes





# Jenkins Continuous Integration for Dev

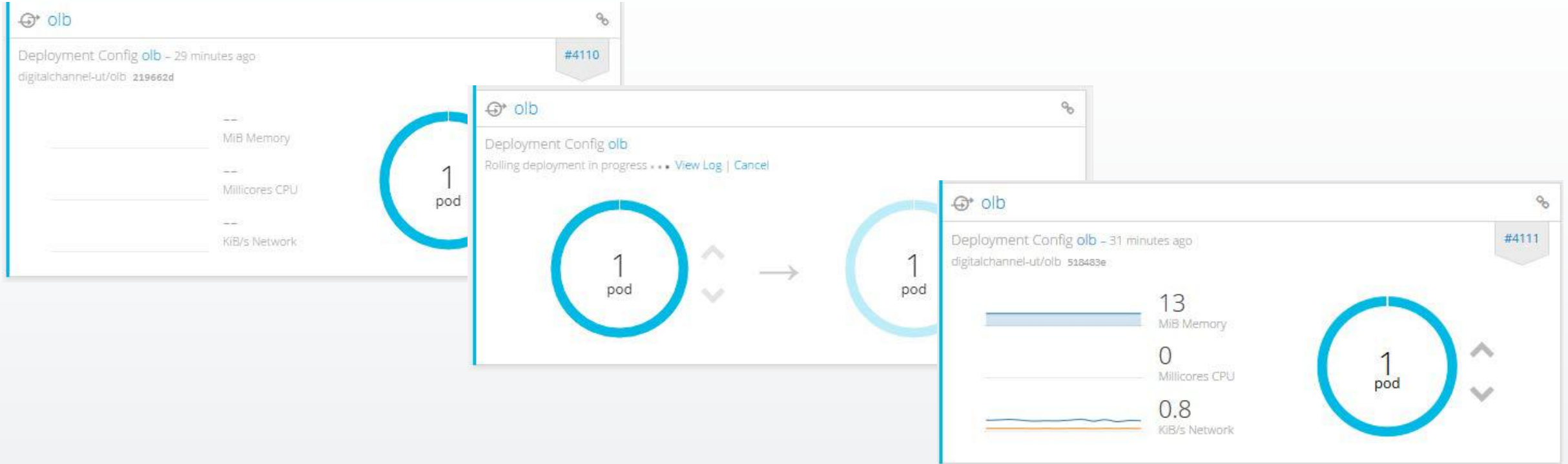


- Went from deploying to lower environments every few weeks to 20+ deployments a day.
- Over 6,000 tests executed on every commit insure quality
- Deploy to OpenShift via rolling deployments
- Separate jobs to push images to higher environments





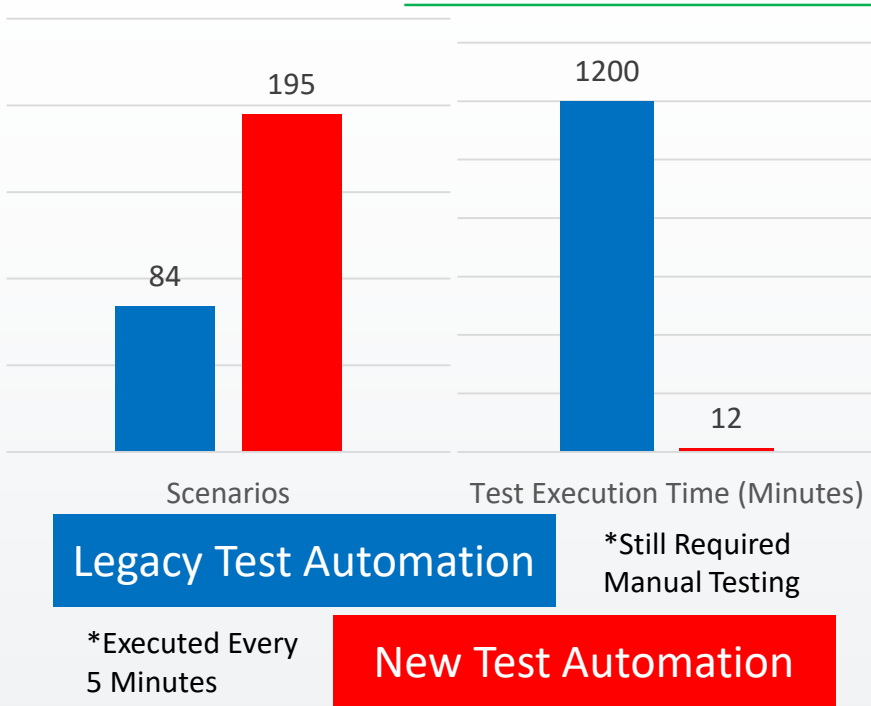
# OpenShift Rolling Deployments



- OpenShift has image triggers, when a new Docker image is pushed into the registry it will perform a rolling deployment



# Early DevOps Success



Bank Acquisition Customer Day One



**10** # of Production Releases to Respond to Customer Input in the first 4 days

**0** # of Defects Resulting in Production Releases

- During highest commit phase Key was identifying 10 defects / day
- Averages roughly 7000 tests executing per build
- 60% defects resolved within a day



**Consistent** Release Process that **Guarantees Versions are Deployed** in a Dependable Automated Manner





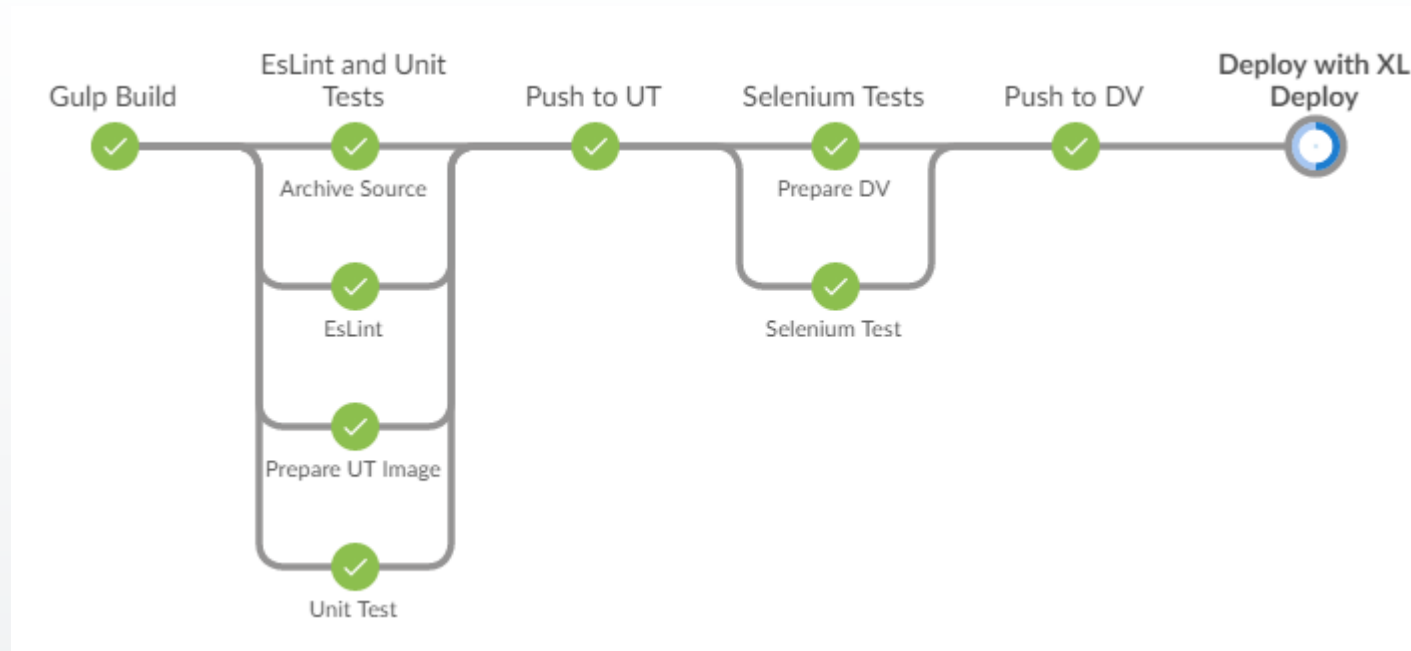
# Maturing Required Changes

---

- Jenkins Pipeline allowed for great CI and delivery velocity
  - Also exposed weaknesses in the system
  - Things accidentally made it to an environment, and from there were pushed to production
- As a bank, needed more audit data
- Needed better rollback abilities
- Needed a full Enterprise Tool



# Updating and going beyond CI Pipeline



- Increased parallelization and tests run
- Built XL Deploy packages, and used them for Deployments



# Better Control of Deployments with XL Release and XL Deploy

Please select a package to Deploy for olb

IBX Release / Package User Input

Please enter the required information below.

**Please select a version to be deployed \***

- Precondition
- Comments (0)
- Attachments (0)

✓ undefined

Applications/OpenShift/ibx00/olb/1.0.2509

Applications/OpenShift/ibx00/olb/1.0.2506

Applications/OpenShift/ibx00/olb/1.0.2504

Applications/OpenShift/ibx00/olb/1.0.2502

Applications/OpenShift/ibx00/olb/1.0.2501

Applications/OpenShift/ibx00/olb/1.0.2498

Applications/OpenShift/ibx00/olb/1.0.2496

Applications/OpenShift/ibx00/olb/1.0.2492

Applications/OpenShift/ibx00/olb/1.0.2491

Applications/OpenShift/ibx00/olb/1.0.2488

Applications/OpenShift/ibx00/olb/1.0.2487

Applications/OpenShift/ibx00/olb/1.0.2485

Applications/OpenShift/ibx00/olb/1.0.2483

Applications/OpenShift/ibx00/olb/1.0.2481

Applications/OpenShift/ibx00/olb/1.0.2479

Applications/OpenShift/ibx00/olb/1.0-may-hotfix.24-prod

Applications/OpenShift/ibx00/olb/1.0.2474

Applications/OpenShift/ibx00/olb/1.0.2472

Applications/OpenShift/ibx00/olb/1.0.2468

Applications/OpenShift/ibx00/olb/1.0.2467

Applications/OpenShift/ibx00/olb/1.0.2466

Applications/OpenShift/ibx00/olb/1.0.2464

Applications/OpenShift/ibx00/olb/1.0.2462

Applications/OpenShift/ibx00/olb/1.0.2459

Applications/OpenShift/ibx00/olb/1.0.2458

**In progress**

Set status text

**Start date**  
16 June 2017  
Friday  
2:24 PM

**Due date**  
16 June 2017  
Friday  
3:24 PM

**Duration**  
0d 1h 0m

**Assigned to** Assign to me

assign

tla00

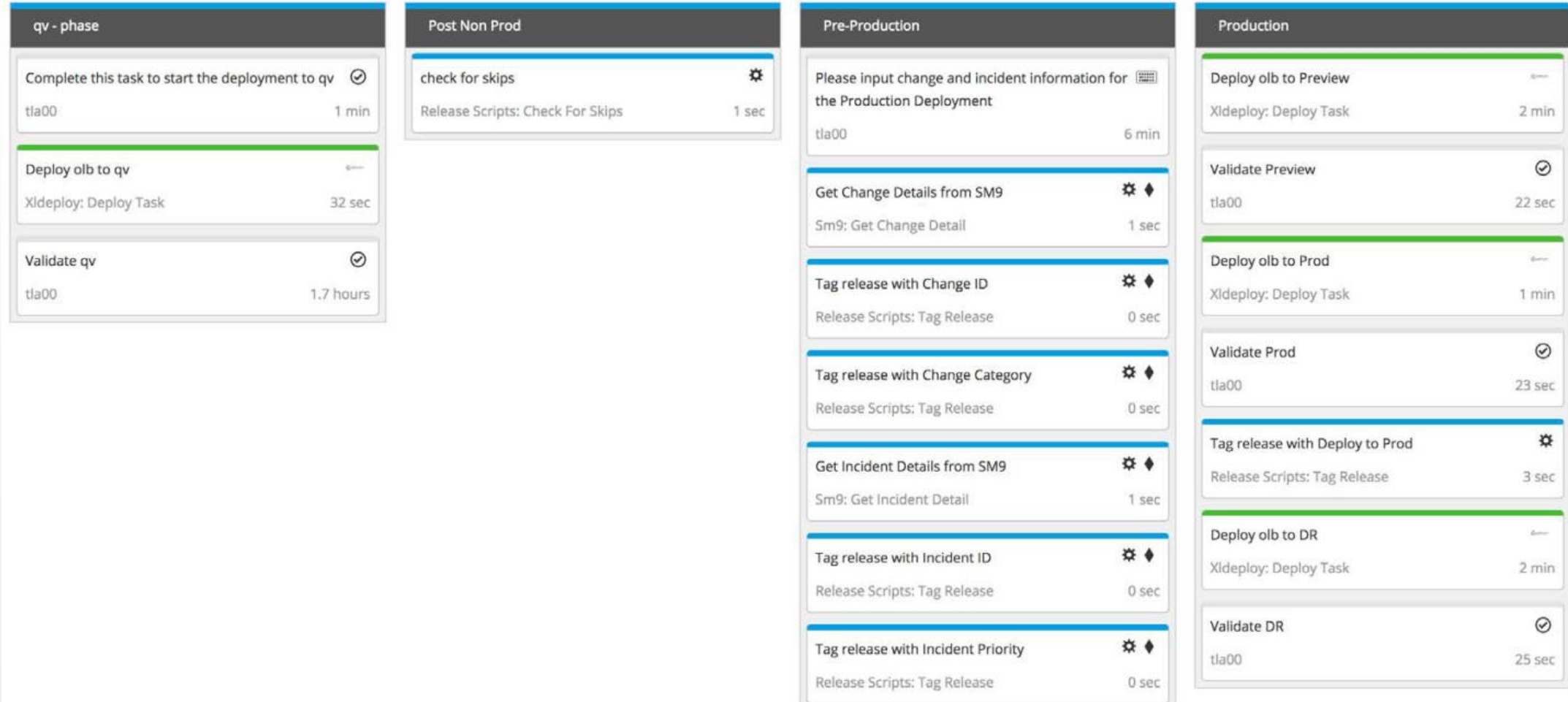
**Watchers** Add me

add user

**Tags**  
Add a tag...



# XL Release Pipeline





# OpenShift XL Deploy plugin

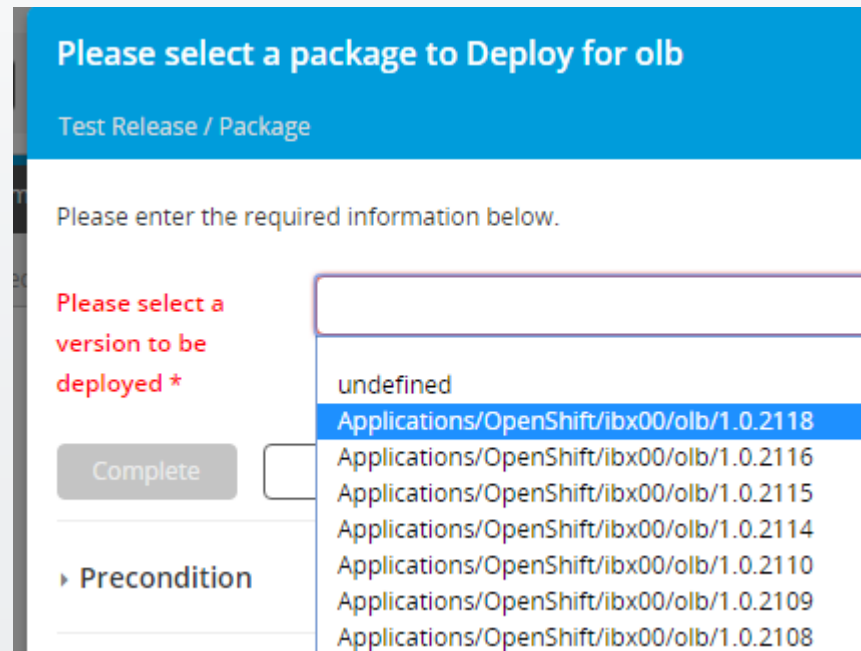
---

- Existing community plugin used “oc new-app” and “oc delete” to perform deployments
- During POC wrote our own plugin from scratch to do deployments
- Because access to source code available, could add new type which performed rolling deployment
- Changes merged into XL Deploy OpenShift plugin version 7.1.0



# Using the Plugin

- Jenkins builds Docker images and pushes into a registry tagged with a unique identifier, build number, commit hash etc.
- XL Deploy package created with this image as the DockerPushApp type







# How the DockerPushApp type works

---

- Issues “docker pull” command to pull tagged image out of the registry
- Retags the image with name of image running in OpenShift
- Pushes image into OpenShift registry
- This triggers OpenShift rolling deployment
- Uses OpenShift API to monitor deployment and report status on completion



# Next Steps

---

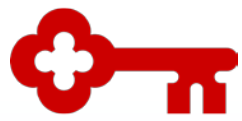
- Update the plugin to allow the deployment of:
  - Deployment configurations
    - Change the number of pods running in each environment
    - Change the Readiness/Liveliness probes
  - Configuration Maps
    - Store data that gets injected into the pod for each environment
  - Secrets
    - Protected Config Maps for passwords and other sensitive material



# Do's and Don't's

---

- Do question "why" each step of your process exists. How can you streamline it?
- Do, and fail. It is how you learn what works and what doesn't.
- Do get the feedback to the developers as soon as possible.
- Don't spend too much time planning the perfect system, it will change as you learn.
- Don't ever think you've mastered something. Always a new and better way to do it.
- Don't forget to have fun.



# Questions

---

Tom Larrow  
@tomlarrow