Dataverse Scaling: Sprint 3 Demo

Students: Michael Clifford, Patrick Dillon, Ryan Morano & Ashwin Pillai

Mentors: Phil Durbin (Harvard), Dan McPherson & Solly Ross (both Red Hat)

Reminder of Project Goals & Scope

- Dataverse was developed as an N-tier web app
  - 1 HTTP server - Glassfish
  - 1 Database - Postgres
  - 1 Search Indexer - Solr
- Collaboration w/ Red Hat moved these components to Docker images
- Our project is to continue this work and create a configuration where Dataverse can scale these components on OpenShift

Stateful Sets

- Stateful sets create persistent identities for pods (a pod, in our case, being a deployed container of Glassfish or PostgreSQL)
- The identities are simply ordinal values appended to the pod name, e.g. glassfish-0, glassfish-1…
- With these identities, we can create primary/secondary or master/slave relationships where pod 0 is the primary or master

Primary/secondary or Master/slave relationships

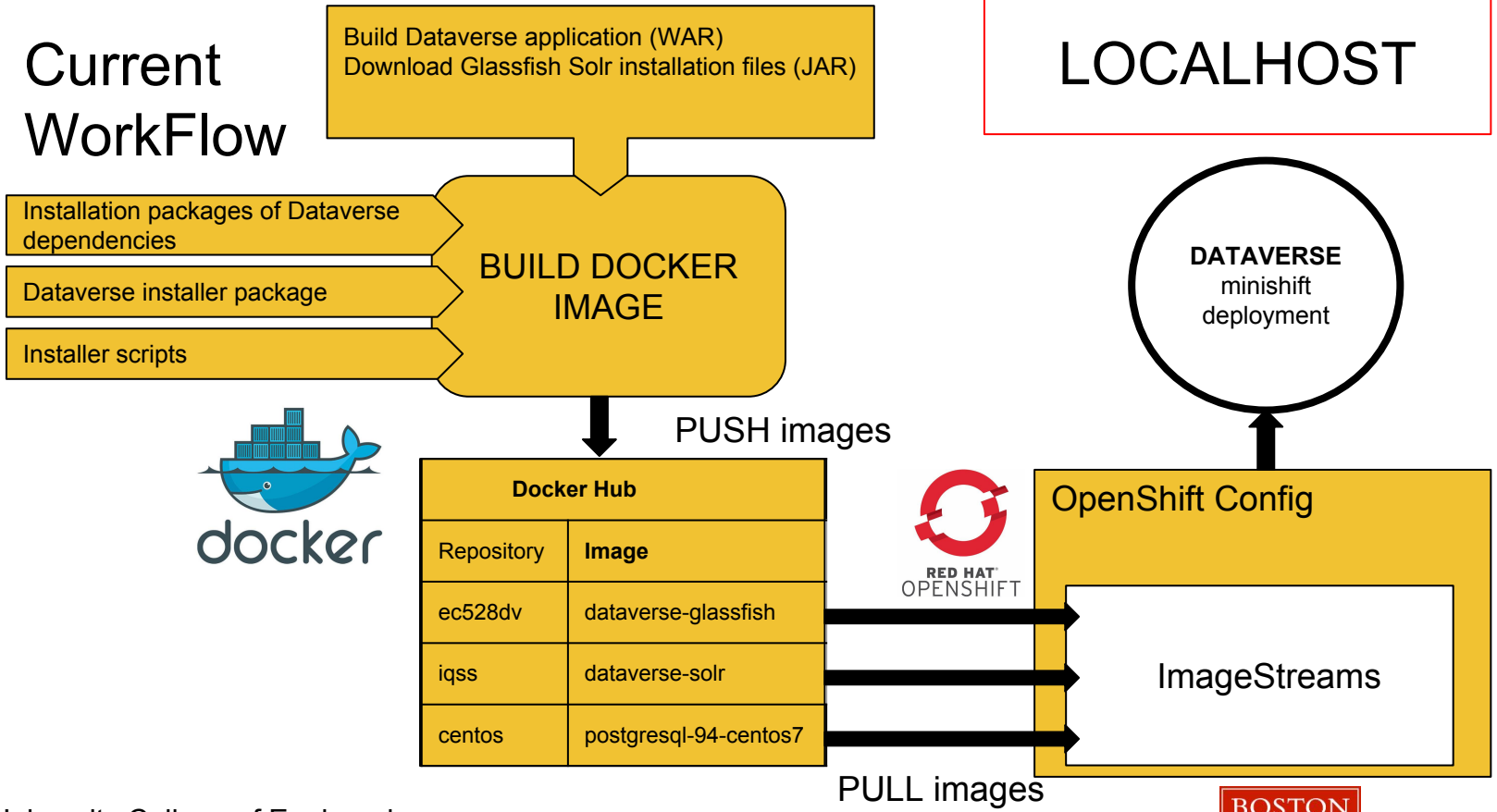
Glassfish: the server executes jobs from a queue; we want only the primary to execute jobs to prevent duplication, but all can serve HTTP requests, because they are stateless

PostgreSQL: in order to maintain consistency, all writes should go to the master, which are then replicated. Our goal is high availability (HA) rather than sharding or another solution.

Work completed

- Developed workflow for updating an entire scalable Dataverse for Openshift
  - Created DockerHub repo to store updated Dataverse images
  - Set up local development machines to update Dataverse files, build updated images and push to docker hub
  - Reconfigured config/openshift.json to pull from new DockerHub image repo
- Updated default.config file to ensure Glassfish talk to master postgresql server
- Updating config/openshift.json to pass appropriate pod name environment variables to the pod
- Update glassfish_setup.sh to restrict jobs to only the 0th pod.
- Updated schedule

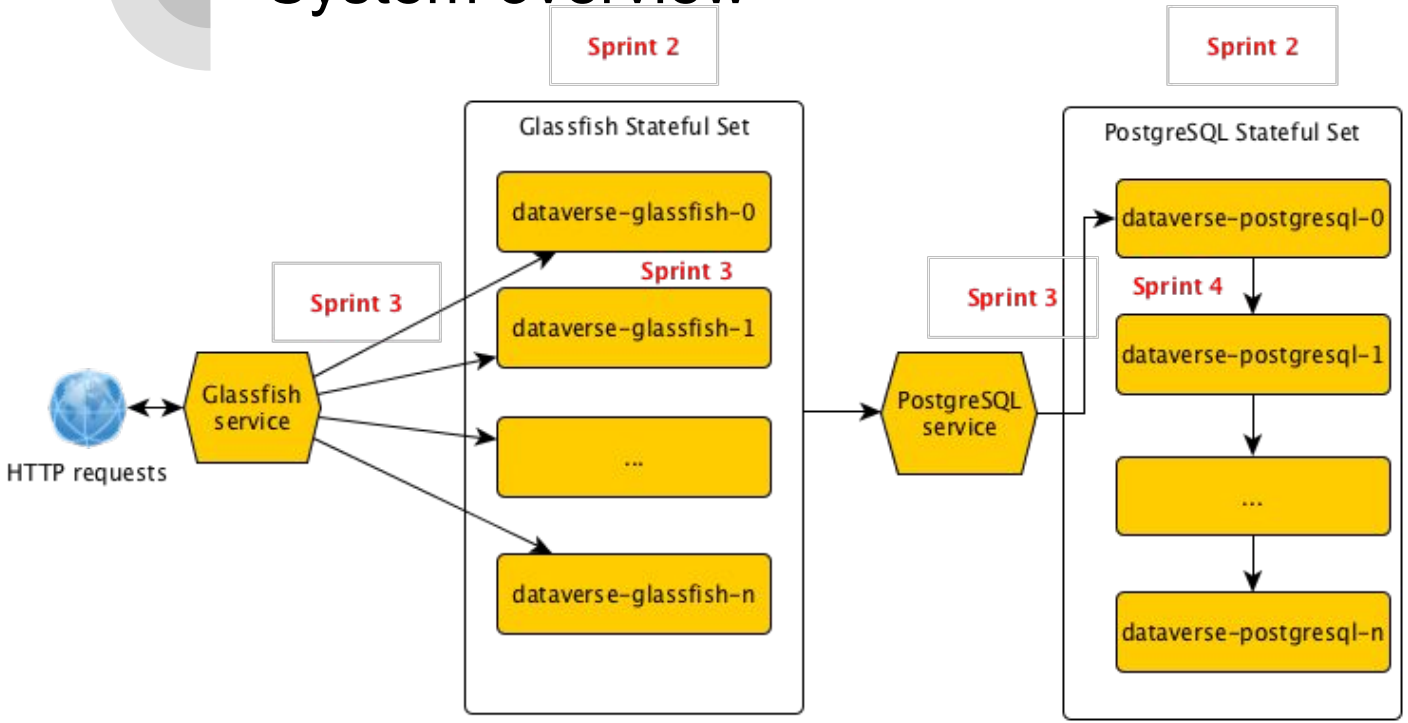
Current WorkFlow



BOSTON UNIVERSITY

DEMO

System overview



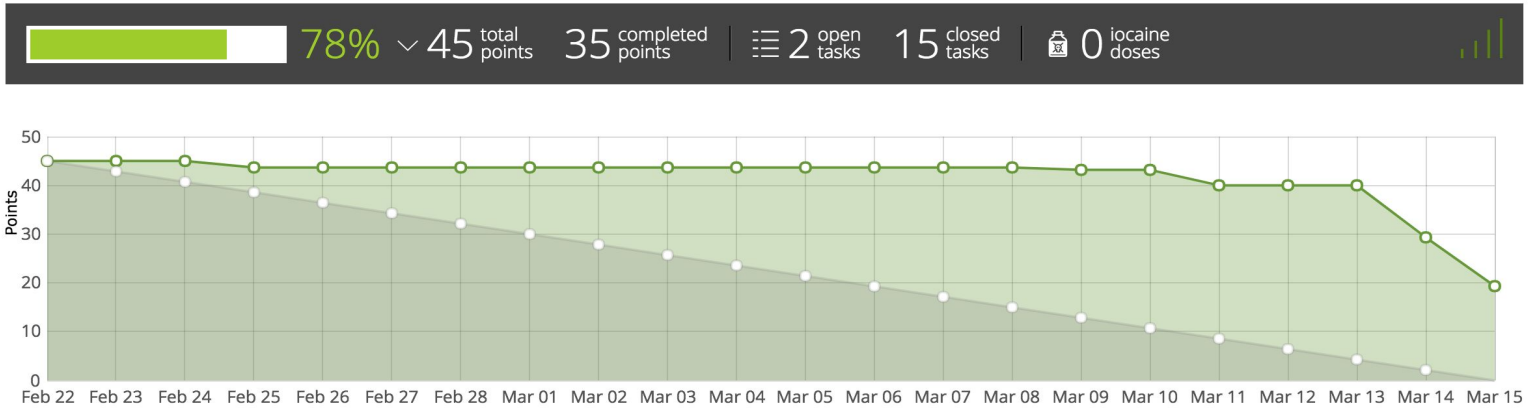
Sprint 2
Create Stateful sets

Sprint 3
Postgres: configure all requests to be sent to master
Glassfish: configure all to serve HTTP requests, only primary dispatches jobs

Sprint 4
Postgres: configure replication to slaves

Sprint Burndown

2018 BUCS528 DATAVERSE SCALE BU CS 528 CLOUD COMPUTING - DEMO 3 22 FEB 2018-15 MAR 2018

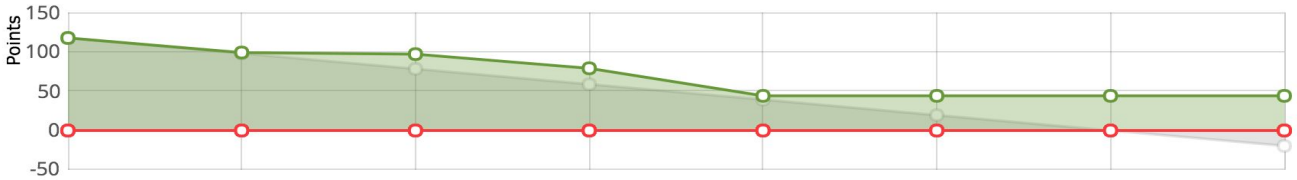


- We spent over 40 man hours on the card “As a developer I want to be able to build and iterate on Docker images” which had an estimate of 1 hour.
- Also the graph seems to be incorrect. We have ten points left (this is shown in top bar) but graph shows 20. It looks more like 50% completed than 78%

Project Burndown & next sprint

2018 BUCS528 DATAVERSE SCALE BACKLOG

64% 117.5 defined points 75.5 closed points 13 points / sprint



- Switch Postgres image from default to one built for high availability (probably Patroni)
- Test Glassfish locally
- Select tool for testing deployment (Apache meter or Jbench)
- Test deployments on MOC

Release Planning

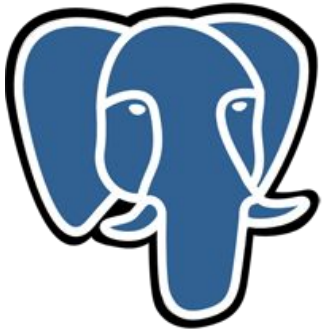
https://tree.taiga.io/project/msdisme-2018-bucs528-template-6/

THANKS!!

The Dataverse Project

Solr

redhat



PostgreSQL