

Dataverse Scaling: Sprint 4 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

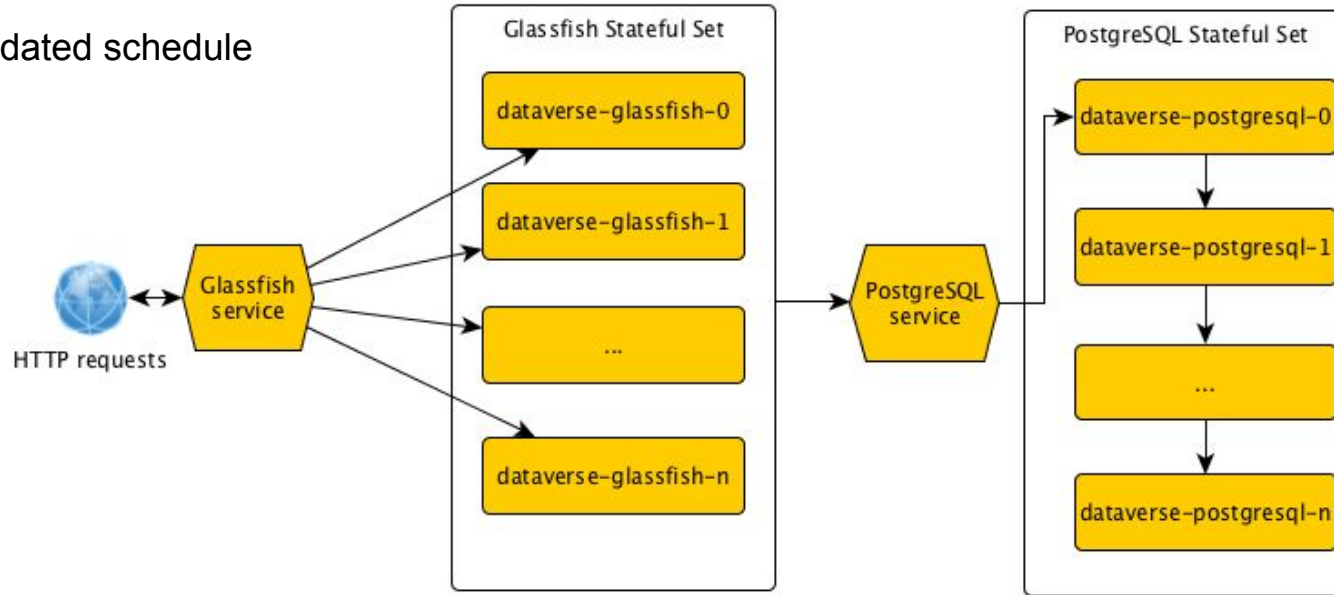
- Dataverse was developed as an N-tier web app
 - 1 HTTP server - Glassfish
 - 1 Database - Postgres
 - 1 Search Indexer - Solr
- Collaboration w/ Red Hat moved these components to Docker images
- Our project is to continue this work and create a configuration where Dataverse can scale these components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





System overview

- Updated schedule





Current WorkFlow

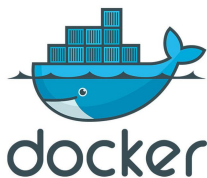
Build Dataverse application (WAR)
Download Glassfish Solr installation files (JAR)

Installation packages of Dataverse
dependencies

Dataverse installer package

Installer scripts

**BUILD DOCKER
IMAGE**



PUSH images

Docker Hub

Repository	Image
ec528dv	dataverse-glassfish
iqss	dataverse-solr
centos	postgresql-94-centos7

patroni



OpenShift Config

ImageStreams

PULL images

LOCALHOST

DATAVERSE
minishift
deployment



Work completed

- Bash script for automating building, pushing and deploying docker images from docker hub if not available locally.
- Switched Centos postgres image with patroni and connected it with glassfish locally.
- Created MOC accounts.
- Deployed Dataverse to MOC's Openshift Container Platform.



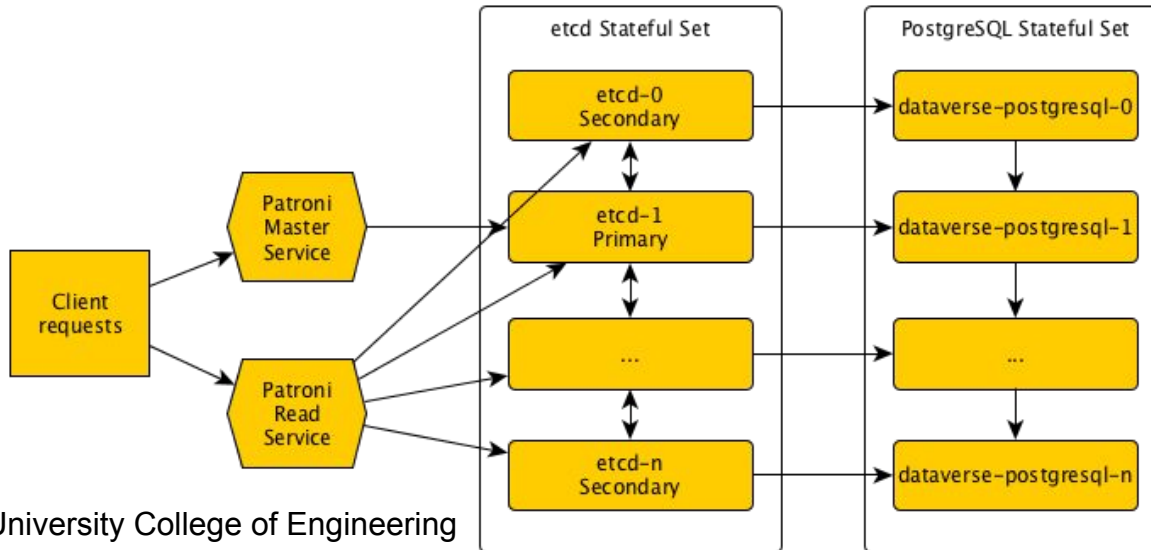
Postgres DB: Switching from Centos to Patroni

- Dataverse was originally setup to use the default Centos Postgres image
- This image would need to be configured for replication when scaling the number of instances
- The Patroni Postgres image is already configured for high-availability on Kubernetes (but not OpenShift, more on this later)



Patroni Postgres Image

Patroni uses etcd key-value store for metadata (e.g. pod is primary/secondary)
The master service always points to the master (in this example, pod 1).
The read service can read from any replica (eventually consistent).





- etcd implements Raft consensus protocol (Ongaro & Ousterhout, 2014)
- Raft is an easier-to-understand version of Paxos (Lamport, 1998)
- Consensus is agreement by agents on values & ordering
- Each pod has timer, when timer expires an election is held for primary
- Primary issues heartbeat notifications which update secondaries and reset their timers
- Allows tolerance of network partitions (e.g. in a 3 node setup with 0 as primary, if 0 is partitioned, 1 and 2 hold election. When 0 comes back online it is brought up to date)



(not fully functional)
DEMO



Patroni status

- We need to do further work to correctly configure services for client and for replication
- Additionally, Patroni is written for Kubernetes
- Unlike Kubernetes, OpenShift does not want containers to run as root
- Patroni wants to create folders and Postgres wants to run as specific user
- Temporarily allowing root, but would like to fix

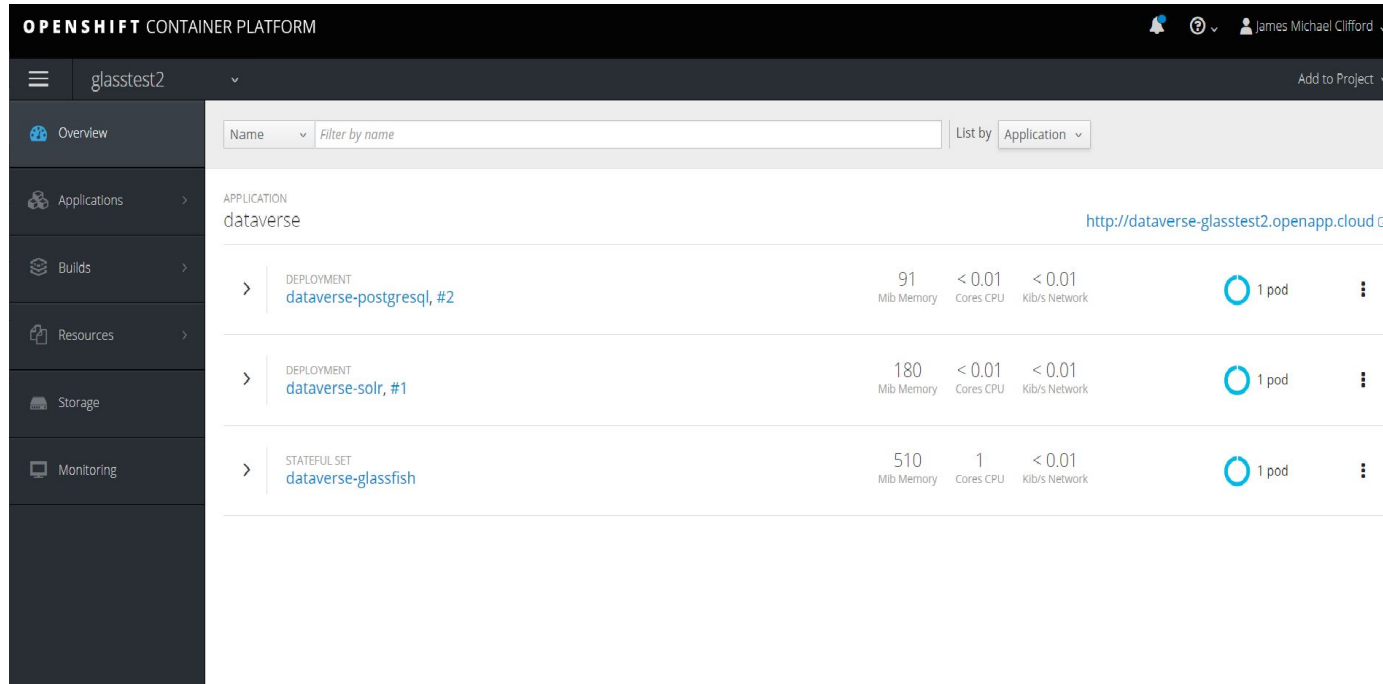


MOC Deployment

- The ultimate goal of our project is to be able to deploy a scalable dataverse on the MOC.
- We have begun to deploy our dataverse images and config.json on the MOC
- Still working on successfully porting working minishift code to MOC openshift.



MOC Deployment



The screenshot displays the OpenShift Container Platform interface for the 'glasstest2' project. The left sidebar contains navigation links for Overview, Applications, Builds, Resources, Storage, and Monitoring. The main content area shows the 'dataverse' application details, including a list of deployments and their resource usage.

OPENSIFT CONTAINER PLATFORM

glasstest2

Overview

Applications

Builds

Resources

Storage

Monitoring

NAME Filter by name List by Application

APPLICATION dataverse <http://dataverse-glasstest2.openapp.cloud>

NAME	DEPLOYMENT	Mib Memory	Cores CPU	Kib/s Network	PODS
>	dataverse-postgresql, #2	91	< 0.01	< 0.01	1 pod
>	dataverse-solr, #1	180	< 0.01	< 0.01	1 pod
>	dataverse-glassfish	510	1	< 0.01	1 pod

Sprint Burndown



2018 BUCS528 DATAVERSE SCALE BU CS 528 CLOUD COMPUTING - DEMO 4 15 MAR 2018-29 MAR 2018



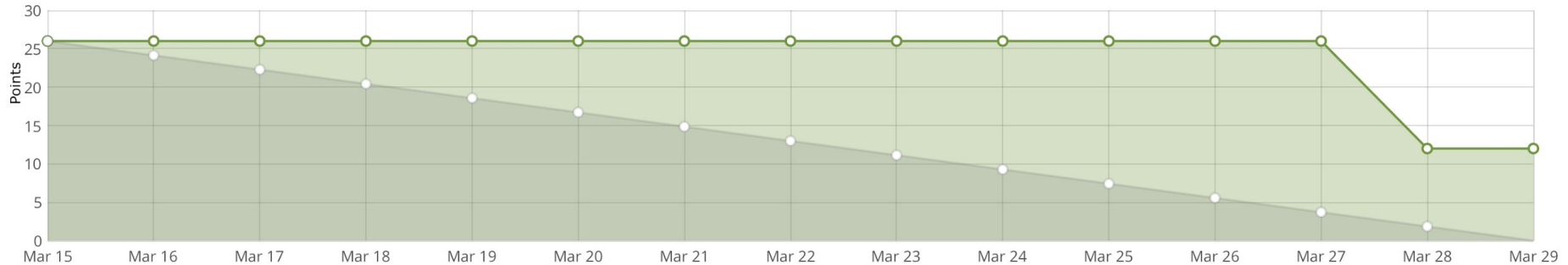
62% \vee 26 total points

16 completed points

1 open tasks

3 closed tasks

0 locaine doses



Project Burndown & next sprint

2018 BUCS528 DATAVERSE SCALE BACKLOG

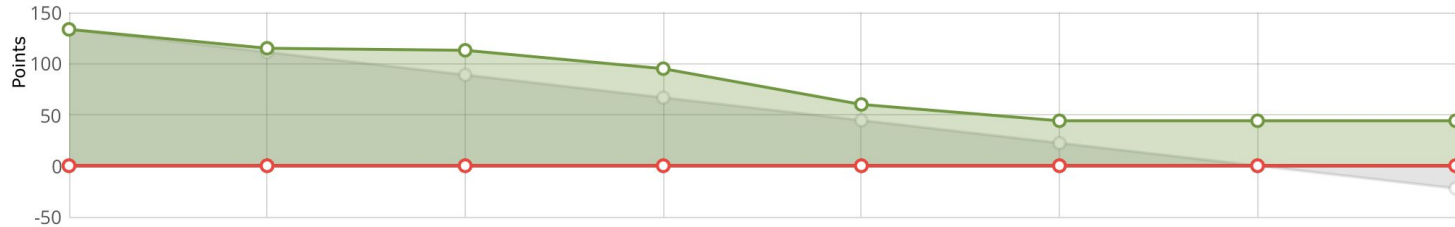


67%

133.5 defined points

89.5 closed points

13 points / sprint



- Patroni deployment
- Pull request
- Select tool for testing deployment (Apache Jmeter or bench)
- Test deployments on MOC



Release Planning



TAIGA

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



THANKS!!



GlassFish

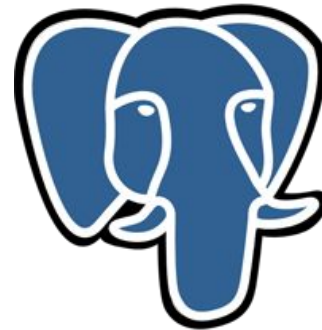
The
Dataverse
Project



Solr 



redhat®



PostgreSQL

Boston University College of Engineering

