

Exploring the self-service model to visualize the results of the ATLAS Machine Learning analysis jobs in BigPanDA with Openshift OKD3

[Ioan-Mihail STAN](#) | Siarhei PADOLSKI | Christopher Jon LEE

PanDA & the BigPanDA

PanDA = **P**roduction **ANd** **D**istributed **A**nalysis system

An abstraction between the ATLAS Distributed Computing infrastructure and tenants

BigPanDA monitoring system \in PanDA

Monitors the entire lifecycle of a task/job submitted to the ATLAS Distributed Computing Infrastructure

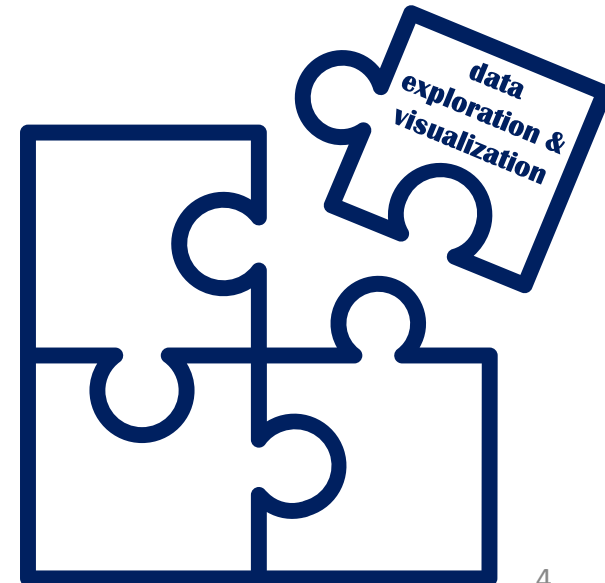


Context

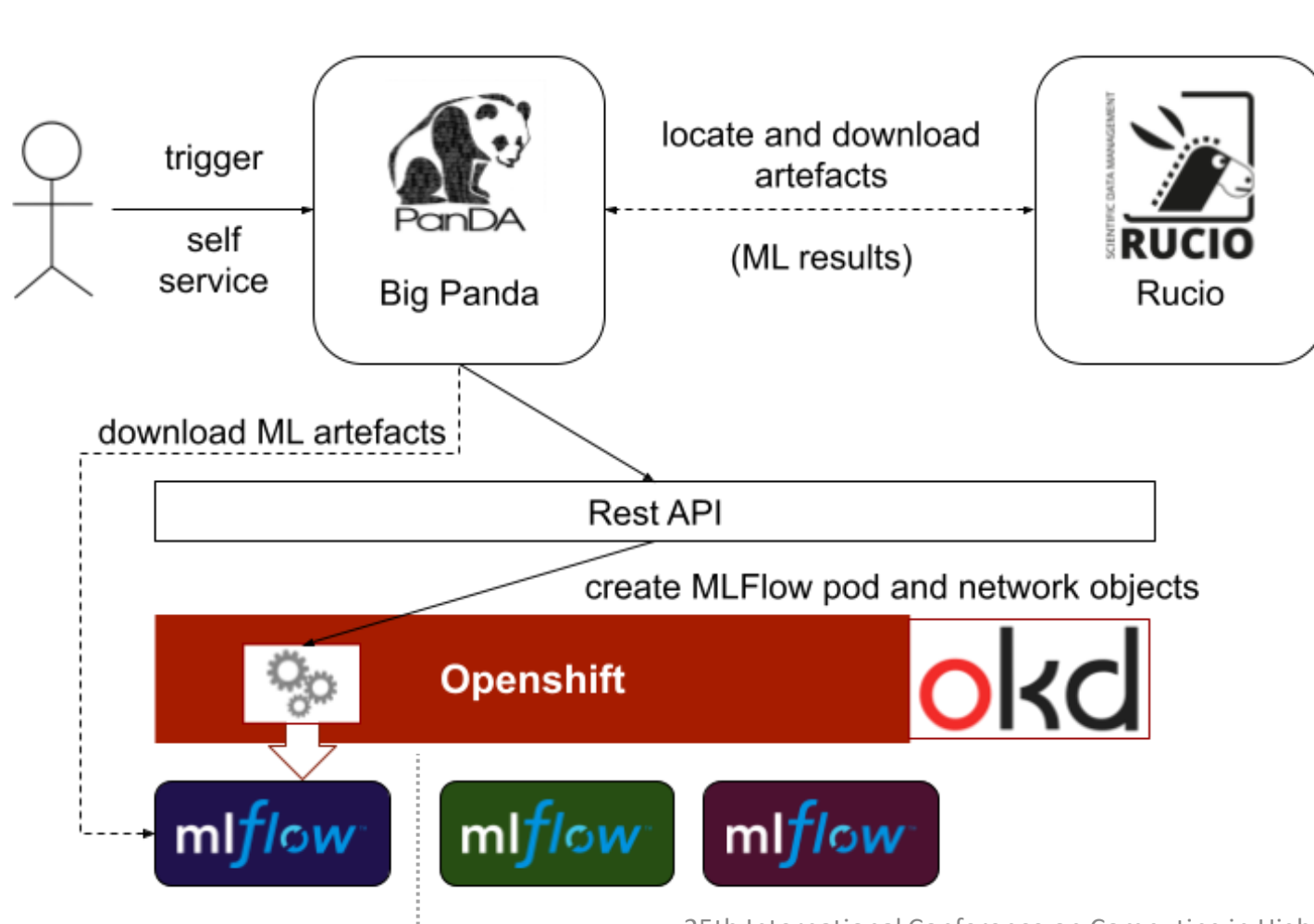
- BigPanDA engineers explored the possibility of concluding a job/task lifecycle with on-demand results visualization services
- A case study has been identified for machine learning jobs that require specialized tools for visualizing and manipulating post-processing data
- Cloud Native models have become a “necessity” for complex, distributed software architectures such as BigPanDA

Our proposal

- Implementing a self-service model for data visualization tools
 - Services catalog available in BigPanDA Dashboard
- Delegating and detaching the visualization functionality to external cloud solutions
 - Generating portable application capable of retrieving ML artefacts and running on remote heterogeneous environments
 - Packaging everything in OCI (Open Container Initiative) compatible images
 - Displaying ML results in browser (as a web service)
- Limiting the exposure of data in time to optimize resource consumption
 - Automatic shutdown and deletion of the web service after 24 hours
 - Dedicated web service for each request



Concept architecture and design

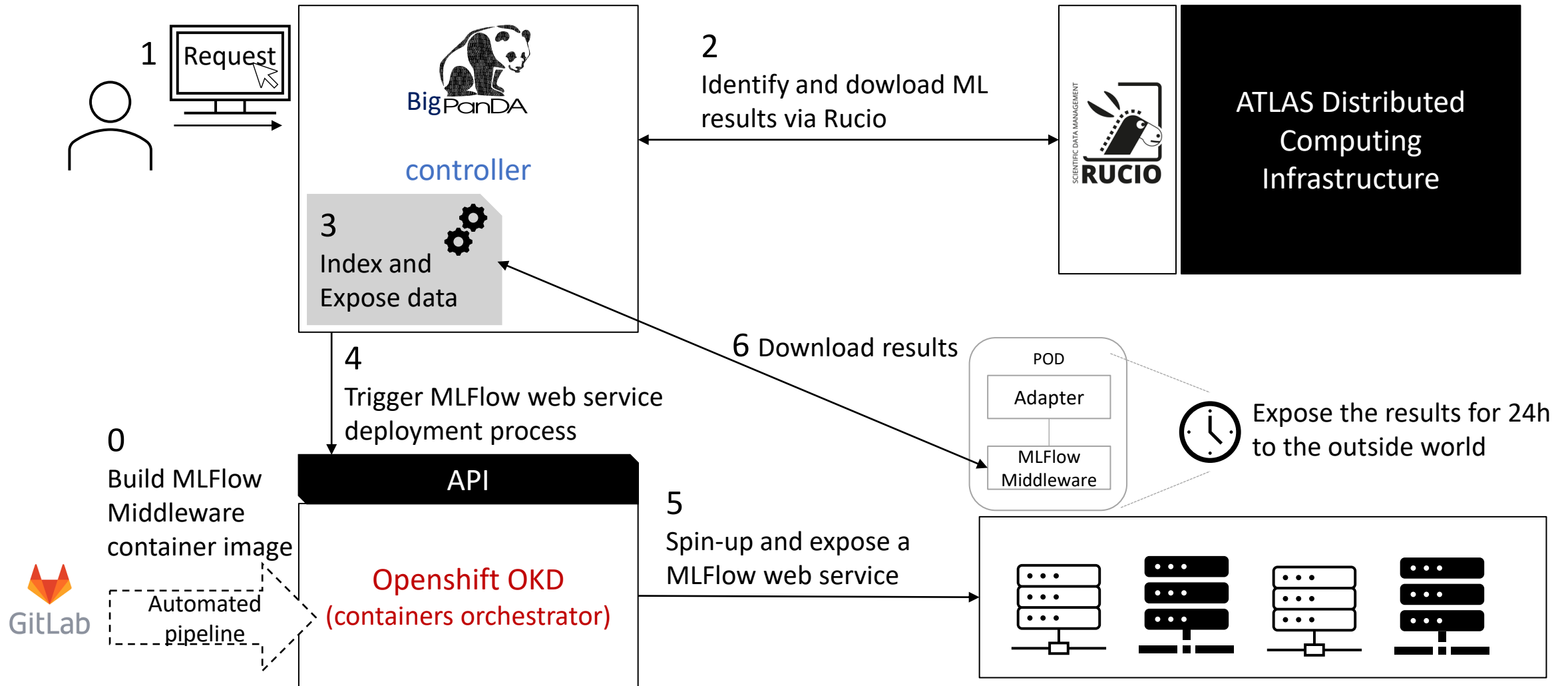


- Openshift OKD 3
 - Kubernetes(K8S) 1.11
 - MLFlow 1.9

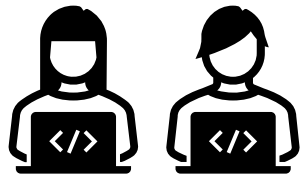
Dedicated infrastructure for running batch jobs

Shared multi-purpose multi-tenant cloud environment

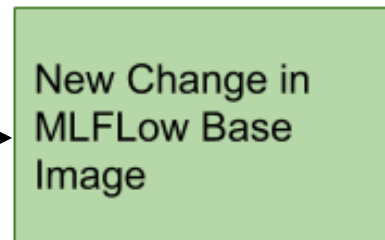
Data Flow



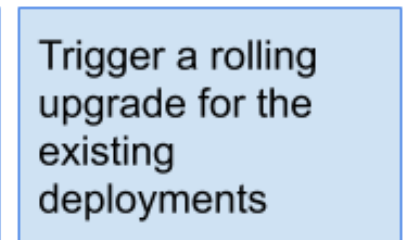
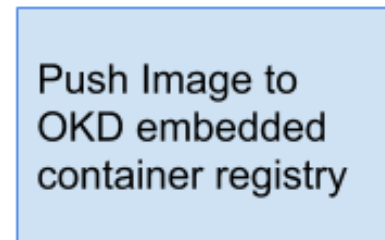
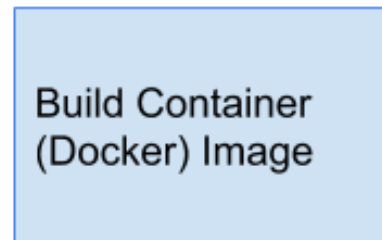
DevOps Practices



>git push



External event



Automated pipelines

- Patches
- Middleware configuration
- Container initialization setup
- Complementary Artefacts

Future improvements and opportunities

- Migration from OKD 3 to OKD 4
 - Pushes the idea of using operators to extend Kubernetes functionality
 - Introduces a newer version of the HA Proxy Ingress Controller with target-rewriting support
 - Simplifies the webservice structure and configuration
 - Currently, we support target-rewriting via a NGINX Adapter with proxy-pass
- MLFlow native tracking routines may provide the mechanics to check the health of the webservice
 - OKD (K8S) supports health probes for services and implements a self-healing mechanism
 - MLFlow Middleware doesn't have a /health endpoint that can be queried

Conclusions

- We experimented with a self-service model for viewing Machine Learning results as an extension of the BigPanDA monitoring service
- We explored a cloud native approach to expanding the BigPanda feature base
- We implemented DevOps practices to automate the construction and updating of data visualization web services

Keywords

- PanDA
- BigPanDA
- Visualization service
- Self-service | On demand service
- Service catalog
- Cloud Native
- Task/Job
- Machine Learning (ML)
- OKD 3/Kubernetes 1.11
- MLFlow 1.9
- DevOps practices
- Automated Pipelines
- Continuous Integration
- Continuous Deployment
- Container Image