



DEPLOYING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING LIFECYCLE ON DIVERSE PLATFORMS USING OPEN SOURCE SOFTWARE STACK

Vivekh Ramakrishnan – *Product Technical Lead, Data and Analytics, Infosys*

Vishnu Sankar – *Senior Architect, Data and Analytics, Infosys*

Santhosh Devarajan – *Technology Lead, Data and Analytics, Infosys*



1.Challenges in setting up an AI/ML platform

Artificial Intelligence has begun providing significant value to organizations in various industries. This will start becoming more apparent in the future as AI solutions become more accessible and hassle-free to implement. However, even with the high levels of interest in leveraging AI and ML solutions, implementation and deployment rate in many organizations is still low. Much of this is because enterprises don't incorporate some realities about AI projects into their thinking.

Challenges that enterprises need to overcome to successfully implement and deploy AI/ML solutions.

Technology & Infrastructure:

- Environment Management & Libraries
 - Multiple libraries and environments needed to execute models. Conflicting requirements impair leveraging common infrastructure
 - Environment requirements for different models are different and executing them on production level needs a robust library management
- Vertical and Horizontal Scaling issues
 - Elastic distribution of workloads with models that can be distributed is restricted to available infrastructure and memory needs.
 - Models which cannot be distributed needs workload split or very high-performance compute.

Model incubation & Building

- Library Access and Installing privileges
 - Access to model libraries and installing them based on needs for incubation is difficult with need for infrastructure resource involvement.

- Takes a lot of time to market due to multiple teams getting involved.
- Collaboration
 - With multiple data scientists working on the same problem with different data sets, for eg. Region-wise tuning of models without the means to collaborate productivity reduces drastically.
- Seamless Actionizing and Measurement
 - Lack of experimentation platform to perform measurement and real world model tuning, impairs the performance of the model in the real world.

2.Challenges in implementing an AI/ML platform implementation

- Do we have a fully automated AI/ML development and collaboration interface for future?
- Will the platform use notebook-based execution or prebuilt models without notebook
- Does the AI/ML model need versioning and parameterization
- How easily can a data scientist communicate the results of a statistical analysis to the business?
- How easily can a data engineer write a code that a test engineer can ensure to run every hour?
- How easily can a ML engineer encapsulate a model iteration which their colleagues can reuse?
- How easily we can reuse AI/ML on existing CICD
- Is there the need of an AI/ML DevOps solution- with automating deployment, scaling, and management of containerized applications

3. How can an AI/ML deployment platform benefit?

- Integrated and abstracted AI/ML DevOps Solutions – hiding all complexities in version control of AI/ML Models, Continuous Integration and Continuous Deployment (CI/CD) with latest or existing toolset
- Framework enables a paradigm shift in how you work with notebook documents – A parameterized notebook for AI/ML Models
- Enabling Notebooks to read or write from multiple places and enabling storing output notebook in a high durability and easy access space in order to provide a reliable pipelines
- Data scientists, Data engineers, DevOps engineers can focus on their work with no other DevOps dependencies.
- Supports automatic deployment (CI/CD) of both scalable and non-scalable algorithms – including the automatic Pods for Parallelism
- Parameterized and Automated support for Scheduling and Triggering the Job – No need for Data scientist or DevOps engineer to understand the complexities of each other's domain
- Quick turnaround – 80 % time saved from a deployment perspective



4.AI/ML Deployment platform

4.1 Building blocks

Data Labs and Unified Semantic Access

- Data Labs for Frictionless Modeling
 - Cloud native Data Labs with all required data synced
 - Capability to associate models to flexible execution containers
 - Full Model Lifecycle from incubation to scoring in a single notebook
 - Real-time micro segmentation
- Comprehensive Data Catalogue and Munging Ecosystem
 - Comprehensive Data Catalogue leveraging tools such as waterline and integrated Munging increases the productivity significantly
- Unified semantic access
 - Unified semantic access framework

allows creating very simple mechanisms to access data from multiple sources and aid in model building.

Elastic and Containerized Execution

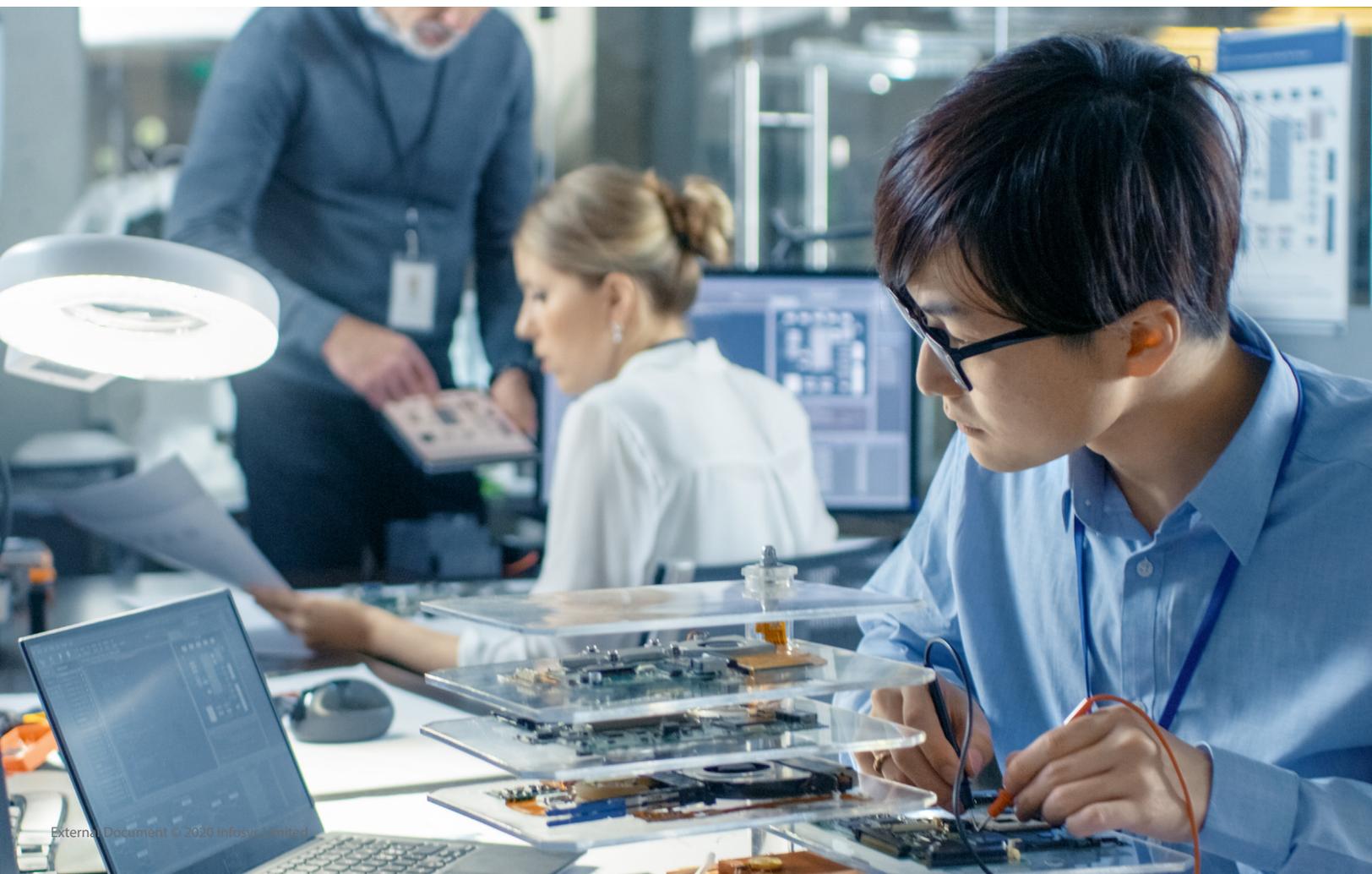
- Elastic and ephemeral execution for models that can be distributed
 - Metadata driven elastic pipelines for model execution for ephemeral processing
 - Parameterized spin up and spin down of instances
- Containerized Execution
 - Creating Containers from the modeling code to bundle the model and the requirements
 - Enabling a pipeline driven execution of Containers in Cloud.
- Incubation to Execution on a Single Notebook – Notebook workflows
 - Leveraging cloud native tools to

create and incubate models in notebook.

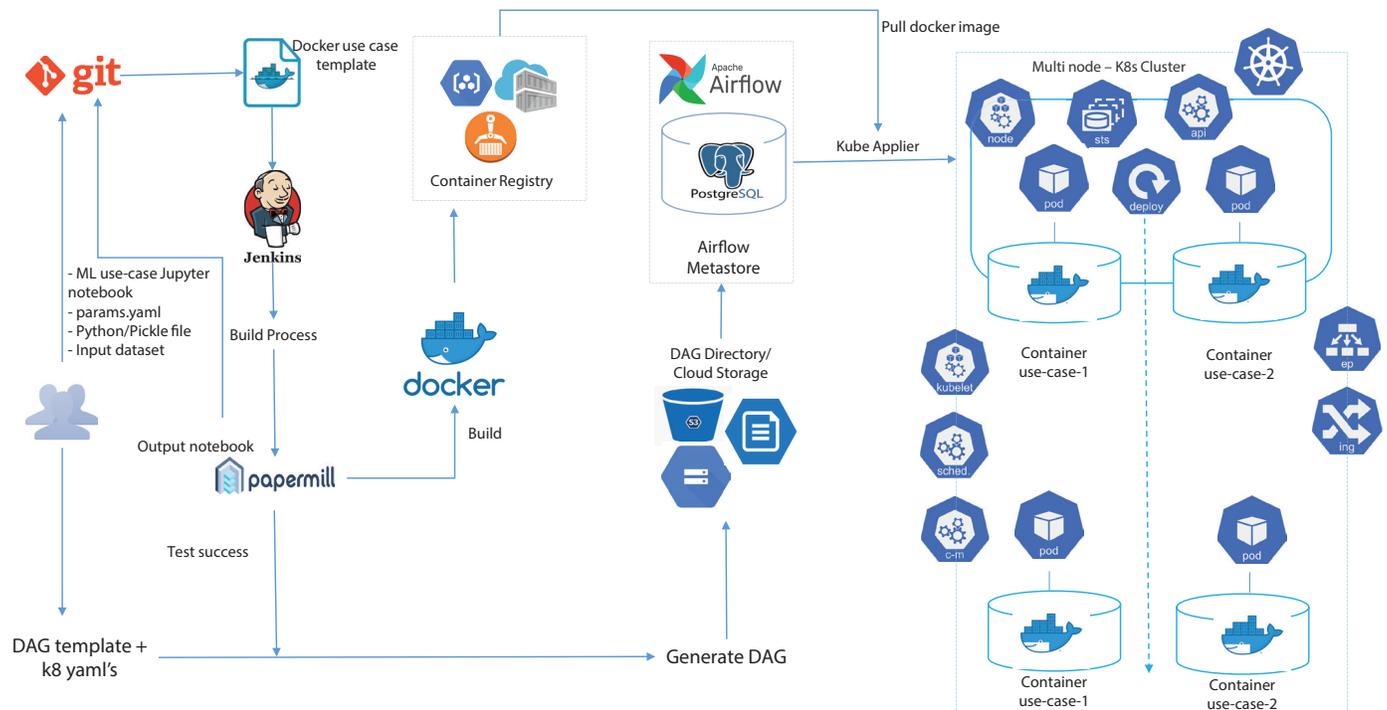
- Leveraging Notebook API to execute the models.

Model Management automated ML Pipeline and Operations

- Modeling kit with Model repository, documentation and notebook templates
 - Repository of models along with their utilization created as notebook templates for citizen data scientists to consume.
 - Once the template is leveraged clear cut CI/CD process to promote the models into production
- Seamless Model to Production
 - Providing CI/CD process to containerize the model and promote to production as containers
 - Integrated testing and data validation for models



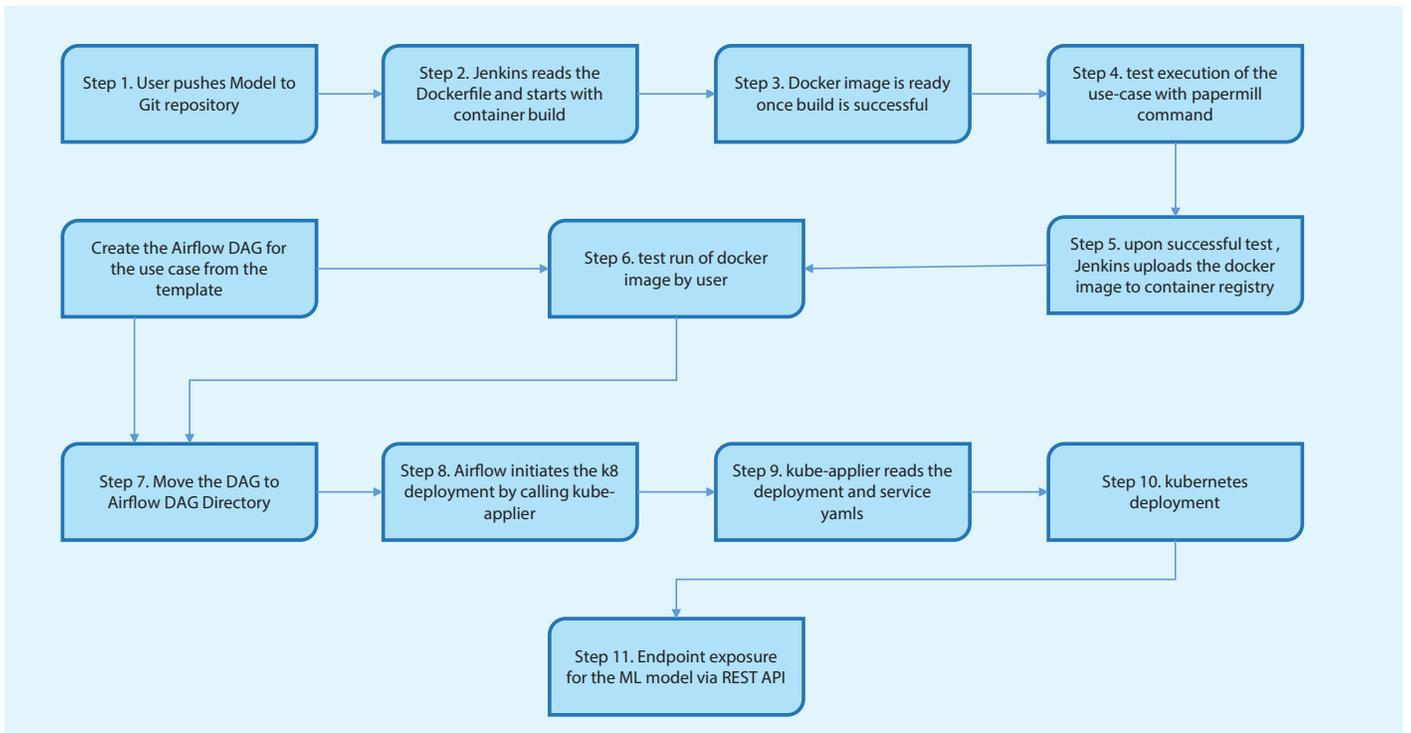
4.2 Deployment Architecture



- Data scientist checks-in the code into the Git repo.
- Jenkinsfile within the repo fires the Jenkins build and docker image of the ML model is created based on the Dockerfile maintained in the repo.
- Upon successful docker image build, papermill command would be fired (within the Dockerfile) to test the Model created with the input dataset.
- Once the output notebook is generated, the entire image is pushed into the container registry. Version history is maintained for the image with the git commit it.
- Airflow DAG is created based on the input template and pushed into the airflow scheduler
- Scheduler picks up the template uploaded and deploys the docker image into kubernetes cluster via kube-applier.
- Kubernetes deployment and service yamls are maintained within the kube-applier.
- ML model is deployed and the test engineer can test the same via a REST API.



4.3 Build & Deploy process flow



4.4 Added benefits

- Versioning history maintained for the code artifacts along with the input dataset used.
- The generated notebook via papermill utility is maintained in Git repo.
- Multiple branches enabled for different environment – sandbox , QA , UAT , PROD
- Containerization (docker) helps to bundle the model as a single entity.
- Containerized entity can be deployed across platforms – Linux/windows/any cloud platform.
- Papermill utility helps to test the ML model before deployment by generating a notebook.
- Container registry helps to maintain versioning of the docker container.
- Scheduling of deployment is via airflow
- Any existing CICD can be used. Need not have a set infra set up for ML deployment.
- Kubernetes help in rolling deployments with zero downtime.
- Kubernetes takes care of auto scaling and de-scaling.
- Service utility in kubernetes helps to expose the ML model as a REST API.
- Multiple environments can be maintained in kubernetes
- Metrics capturing of the ML model becomes easier with kubernetes monitoring tools.
- Application state always maintained by kubernetes. Never crashes.
- Data scientist's role is limited to the development and code merge in Git. Rest of the build and deployment is fully automated
- Python library and environment management is taken care by Docker.
- Scheduled and controlled deployment via airflow.
- Every step in build and deployment can be tracked at any given point of time.



For more information, contact askus@infosys.com



© 2020 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.