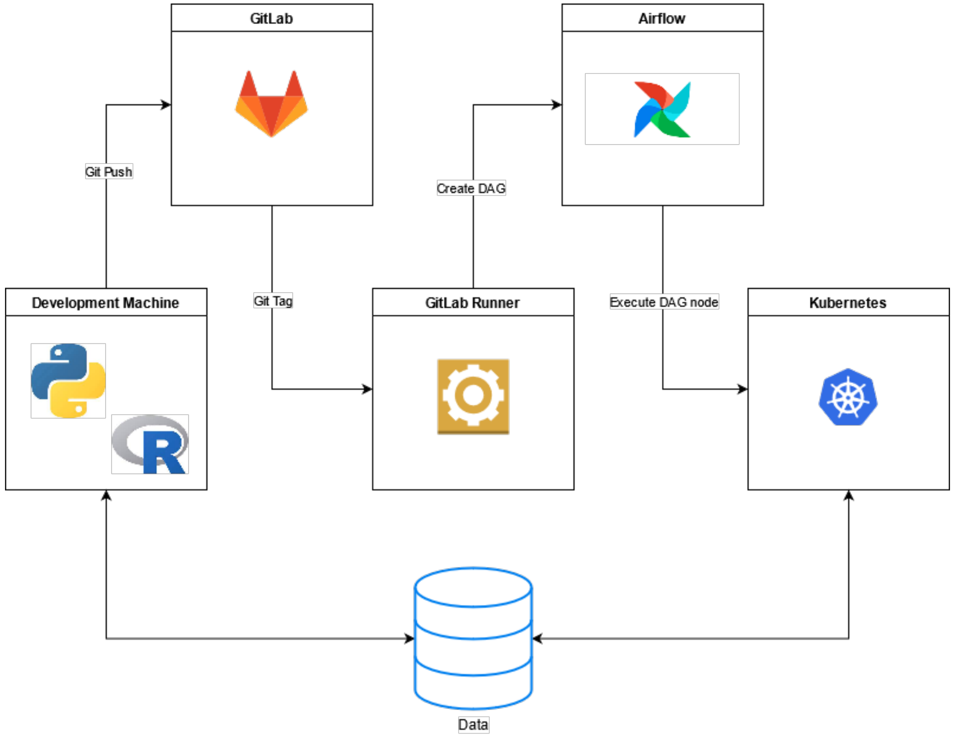
Australian Government
Australian Taxation Office

# MLOps in the ATO

James Beck

# Current Solution

- End-to-end pipeline for model deployment.
- Uses open source software such as Apache Airflow and Kubernetes.
- Largely automated deployments allow for rapid experimentation and reduced lead times.
- Automated pipelines through GitLab.
- Allows for any model type from R and python.
- Models are structured as directed acyclic graphs (DAGs) and packaged into containers.
- Airflow allows for scheduled execution.



“

With Machine Learning Model Operationalization Management (MLOps), we want to provide an end-to-end machine learning development process to design, build and manage reproducible, testable, and evolvable ML-powered software.

”

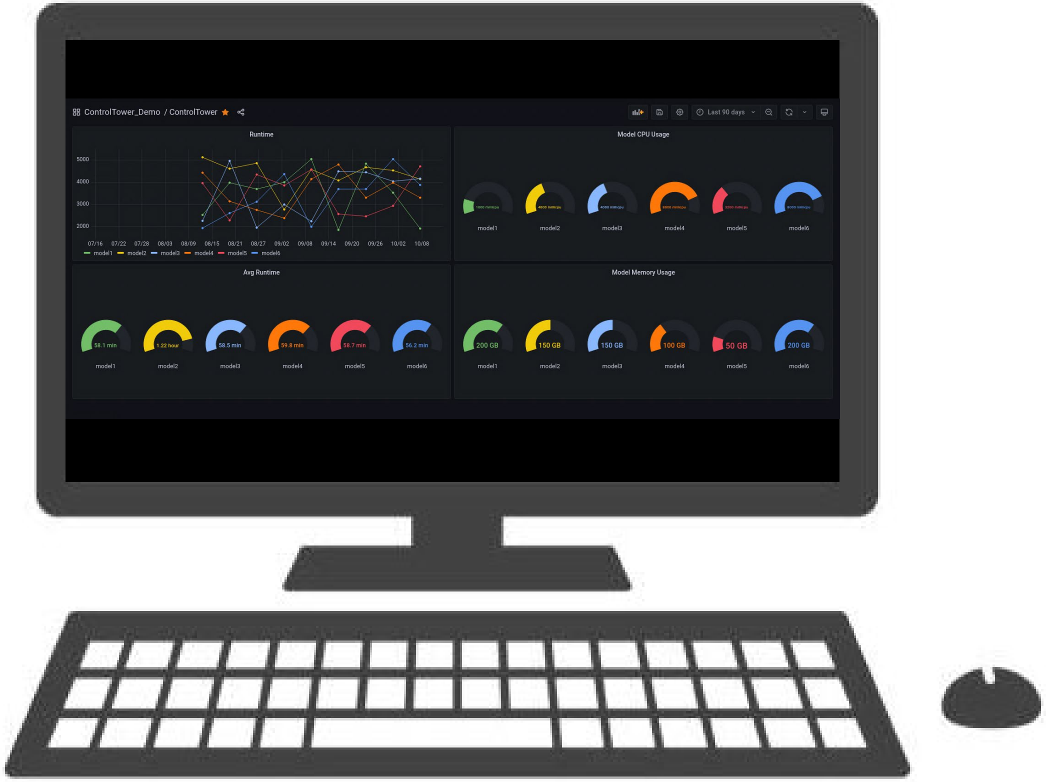
ml-ops.org

# MLOps Principles

| | Monitoring | Versioning | Reproducibility | Experiment Tracking | Testing |
|---|---|---|---|---|---|
| Problem | Models can run often and output a lot of data that needs to be analysed quickly.<br>Want to understand:<br>• Model Drift<br>• Infrastructure Utilization<br>• Model Integrity<br>• Catalog of models | Due to rapid experimentation and changing data, models can iterate quickly.<br>Versioning is needed from both a governance and engineering perspective | As model training often relies on complex statistical and probabilistic algorithms models can be hard to reproduce.<br>Also need to be able to reproduce model inferences for audit purposes. | Many different hyperparameters and configuration to tune when training models. | There are many assumptions made when developing a model that may not be regularly tested. |
| Solution | A central view of all models with key statistics highlighted. | A consistent system for tracking model versions. | The ability to rerun the model training and reproduce an equivalent model. | A consistent system for tracking the model experimentation process. | A system for testing assumptions and assertion throughout the model lifecycle. |
| What the ATO is doing | • A custom built metric aggregation and visualization solution called Control Tower | • Implementation of the open source tool MLFlow<br>• GitLab | • Implementation of the open source tool MLFlow<br>• Best Practice guide | • Implementation of the open source tool MLFlow | • Best Practice guide<br>• Unit Test templates<br>• Automated testing with GitLab |

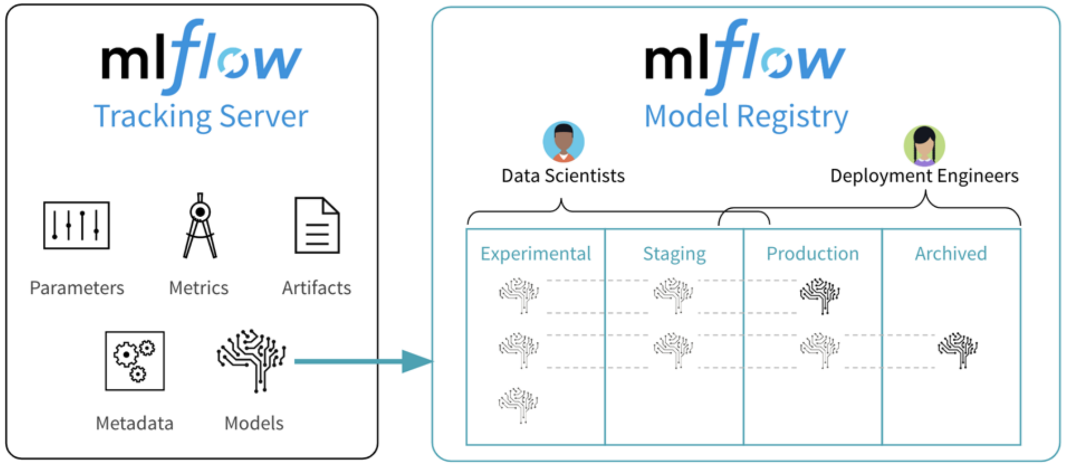
# Control Tower

- Serves multiple monitoring use cases at various levels of the organization.
- Acts as model catalog and metadata store for executives.
- Model metrics such as model training time and score are sent via an API to the Control Tower.
- The metrics are aggregated and stored in a database
- An API can be used to retrieve the metrics so they can be visualized in a series of dashboards.
- Allows for slicing and dicing of data according to metadata
- These metrics can also be used to trigger pipelines such as automated model retaining based on concept drift.



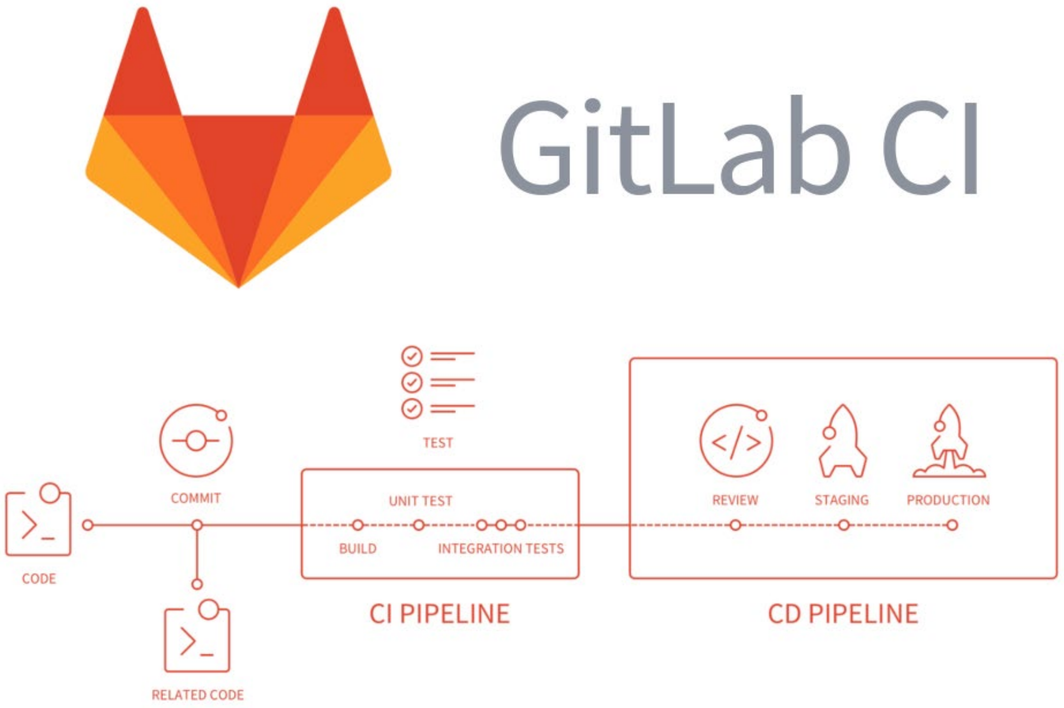
# MLFlow

- MLFlow is an open source tool that contains various components to assist the machine learning workflow.
- MLFlow Tracking allows the user to log the parameters, metrics and artifacts on the model during training. This can be used to choose the best model during cross validation and to provide audibility to the model selection process..
- MLFlow Models provides a consistent interface for models created by a wide range of machine learning frameworks such as Tensorflow, SKLearn and PyTorch. It also supports custom models that don't use standard libraries through the use of a wrapper function.
- MLFlow Model Registry allows for the versioning, staging, inference and serving of MLFlow Models.



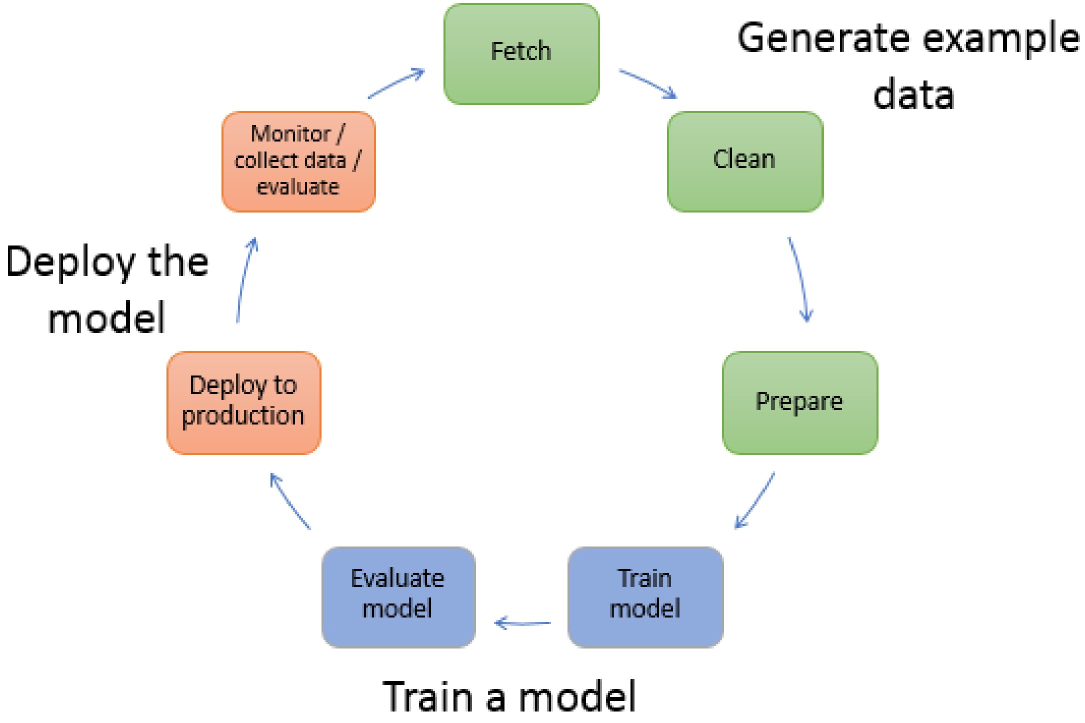
# GitLab

- GitLab is a git repository manager and DevOps lifecycle tool.
- We make use of both the code management and CI/CD (continuous integration and continuous deployment).
- Using CICD we have automated model deployment and testing.
- Allows for validating project structures and model compliance automatically.



# ML Testing

- Testing machine learning pipelines can be complex as there are many distinct steps that run at different times and frequencies.
- The standard categorises of tests include Features and Data Tests, Model Development Tests and ML Infrastructure Tests.
- Google have fantastic paper on this topic: [The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction]().
- It can be difficult to automate the execution of these tests at an infrastructure level as it requires "hooks" into the data science code.



Thank you for listening

Questions and Discussion