

DevOps for AML

How to bring your Data Science project to production

Lorea Arrizabalaga
TSP AA & AI



What is DevOps?

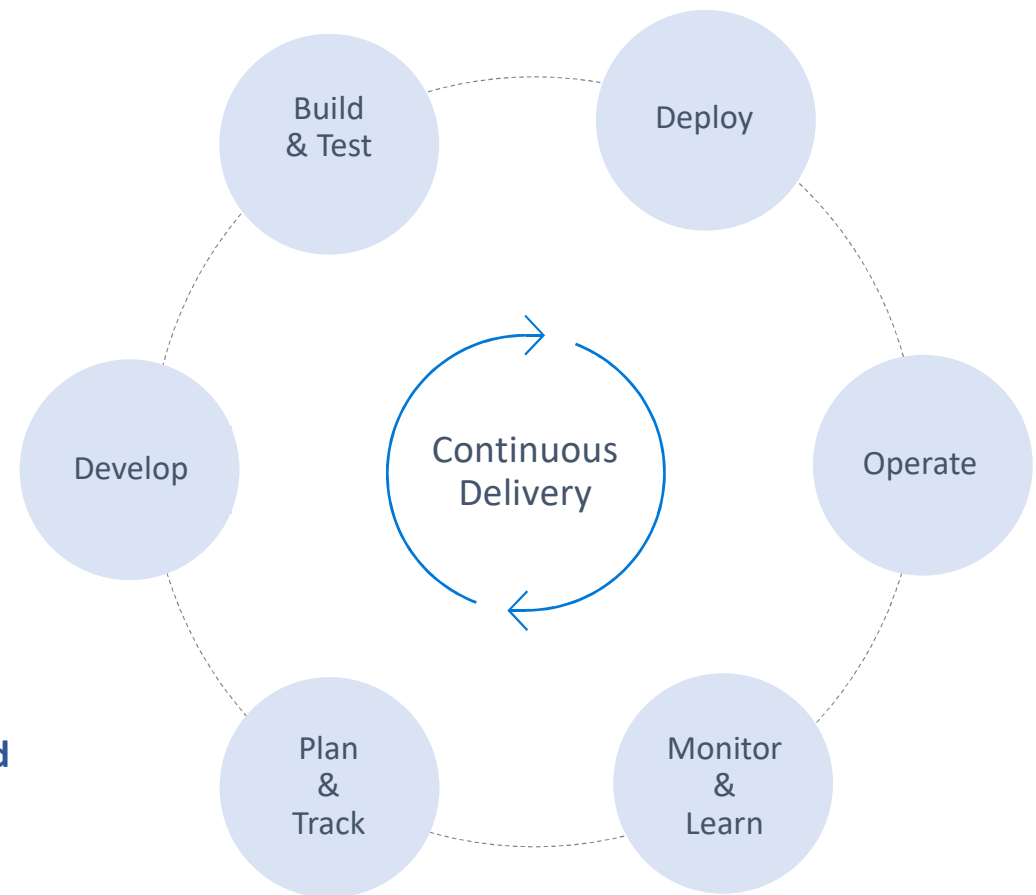
People. Process. Products.



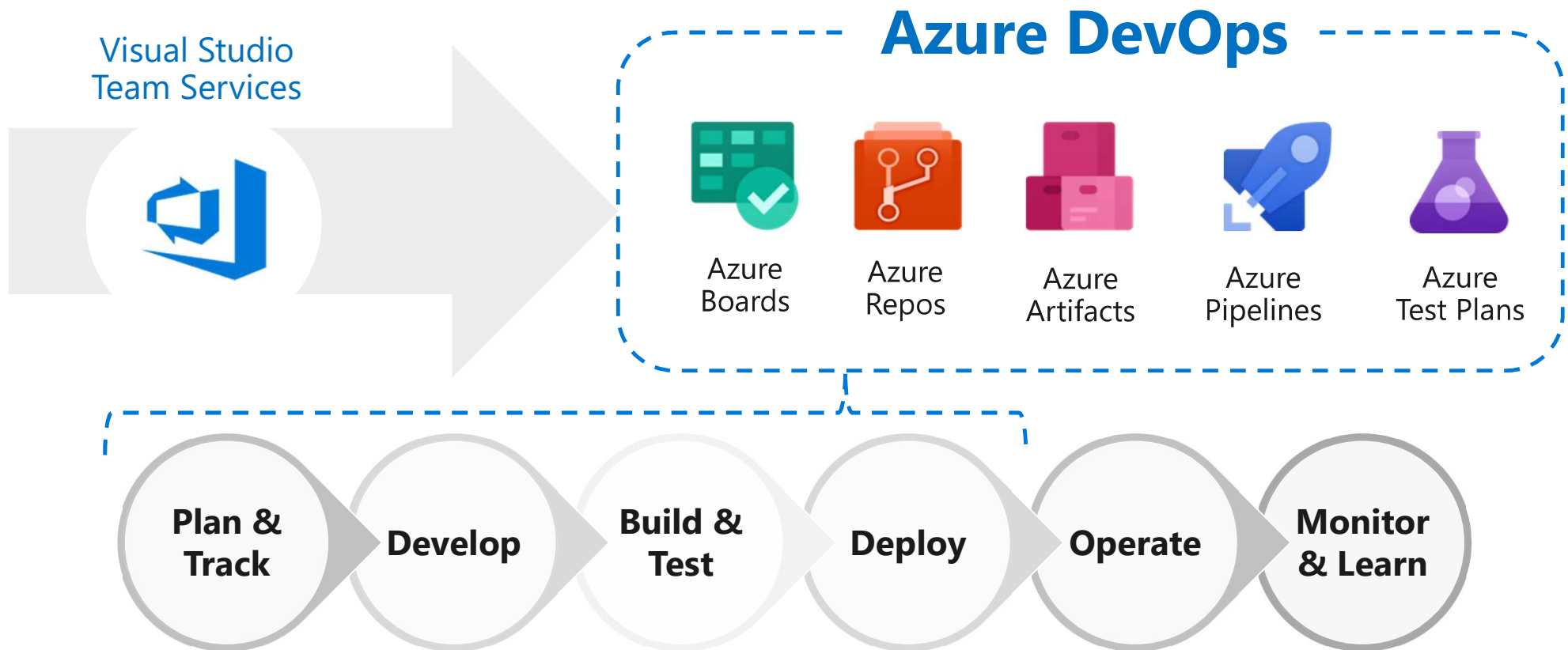
“ Working towards a common goal enabling the **fast flow of planned work into production** while achieving world-class stability, reliability, availability, and security ”

<https://devopsinstitute.com/>

DevOps practice is one evolution to change how IT organizations are delivering core technology services and products to support the business' digital opportunities



Azure DevOps... the evolution of Visual Studio Team Services



DevOps at Microsoft



Azure DevOps is the toolchain of choice for Microsoft engineering with over 80,000 internal users

 <https://aka.ms/DevOpsAtMicrosoft>

372k

Code reviews per month

4.4m

Builds per month

5m

Work items viewed per day

218k

Git code commits per month

500m

Test executions per day

500k

Work items updated per day

42,000

Deployments per day

What does DevOps for AI mean?

How DevOps for AI is different?

Challenges of an AI development process

- **Model quality** highly **correlated to the quality of the training data**
- **Model quality** may **degrade overtime** due to:
 - Concept drift: statistical properties of the data
 - External data dependencies change
- **Model bug** fixing is a lot **more complex** than in traditional software development

Lead to augmented needs for:

- Scripting and **automating** the build/training model process
- **Testability** of code & model behavior
- Richer **monitoring** and customer **feedback loops**

DevOps for AI requirements

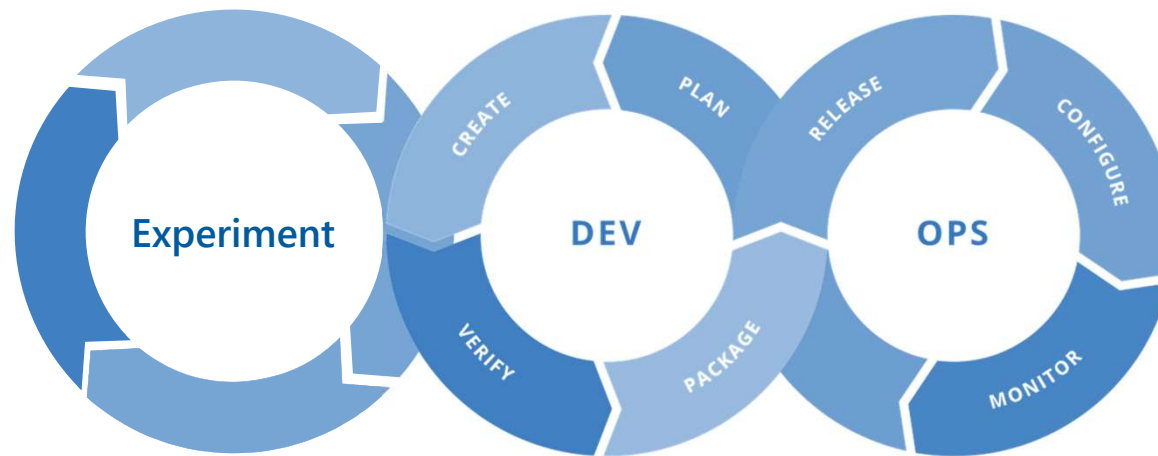
DevOps is the standard way to manage application lifecycles through a pipeline of code, test, build, deploy (CI/CD cycle) to continuously deliver value to end users.

Infusing AI into this lifecycle brings new challenges: a more complex process that requires more flexibility than traditional software development.

A CI/CD solution for AI requires supporting:

- **Reproducibility of data** → model
- **Validation of model** → (does it meet quality bar, A/B comparison)
- **Storage, versioning** → (track lineage and evolution of model over time)
- **Deployment, tracking, data collection** → (across intelligent cloud + edge)
- **Complex Security** → (project security, deployment security)
- **Feedback Loop** → (model refinement or adaptation to changing environment)

AI DevOps lifecycle



Experiment

Data Acquisition
Business Understanding
Initial Modeling

Develop

Modeling+ Testing
Continuous Integration
Continuous Deployment

Operate

Continuous Delivery
Data Feedback Loop
System + Model Monitoring

Key Goals of DevOps for AI

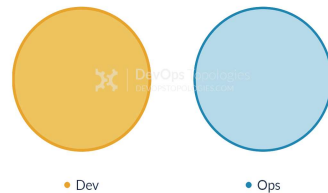
- **Produce repeatable experiments**
 - Repeatability of model creation & behavior
 - Evaluation of model predictions
- **Manage model lifecycle**
 - Different model versions, files, dependencies
 - Operationalization of the model
 - Monitoring of training, CI/CD, scoring pipelines
- **Keeping up with research**

What doesn't work in DevOps

Anti-Type A: Dev and Ops Silos

This is the classic 'throw it over the wall' split between Dev and Ops. It means that story points can be claimed early (DONE means 'feature-complete', but not working in Production), and software operability suffers because Devs do not have enough context for operational features and Ops folks do not have time or inclination to engage Devs in order to fix the problems before the software goes live.

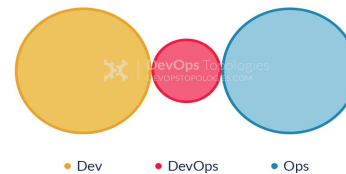
We likely all know this topology is bad, but I think there are actually worse topologies; at least with Anti-Type A (Dev and Ops Silos), we know there is a problem.



Anti-Type B: DevOps Team Silo

The DevOps Team Silo (Anti-Type B) typically results from a manager or exec deciding that they 'need a bit of this DevOps thing' and starting a 'DevOps team' (probably full of people known as 'a DevOps'). The members of the DevOps team quickly form another silo, keeping Dev and Ops further apart than ever as they defend their corner, skills, and toolset from the 'clueless Devs' and 'dinosaur Ops' people.

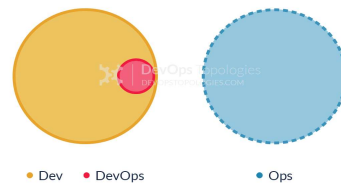
The only situation where a separate DevOps silo really makes sense is when the team is temporary, lasting less than (say) 12 or 18 months, with the express purpose of bringing Dev and Ops closer together, and with a clear mandate to make the DevOps team superfluous after that time; this becomes what I have called a **Type 5 DevOps Topology**.



Anti-Type C: Dev Don't Need Ops

This topology is borne of a combination of naivety and arrogance from developers and development managers, particularly when starting on new projects or systems. Assuming that Ops is now a thing of the past ("we have the Cloud now, right?"), the developers wildly underestimate the complexity and importance of operational skills and activities, and believe that they can do without them, or just cover them in spare hours.

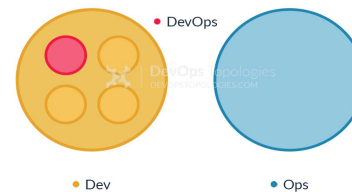
Such an Anti-Type C DevOps topology will probably end up needing either a **Type 3 (Ops as Iaas)** or a **Type 4 (DevOps-as-a-Service)** topology when their software becomes more involved and operational activities start to swamp 'development' (aka coding) time. If only such teams recognised the importance of Operations as a discipline as important and valuable as software development, they would be able to avoid much pain and unnecessary (and quite basic) operational mistakes.



Anti-Type D: DevOps as Tools Team

In order to "become DevOps" without losing current dev teams velocity (read delivery of functional stories), a DevOps team is set up to work on the tooling required for deployment pipelines, configuration management, environment management, etc. Meanwhile Ops folks continue to work in isolation and Dev teams continue to throw them applications "over the wall".

Although the outcomes of this dedicated team can be beneficial in terms of an improved tool chain, its impact is limited. The fundamental problem of lack of early Ops involvement and collaboration in the application development lifecycle remains unchanged.



<https://web.devopstopologies.com/>

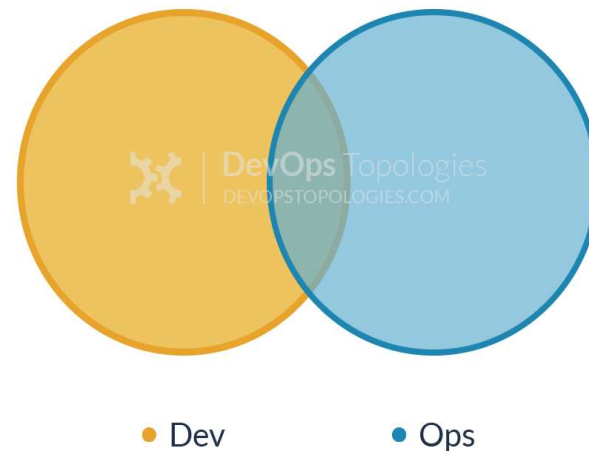
What works in DevOps

Type 1: Dev and Ops Collaboration

This is the 'promised land' of DevOps: smooth collaboration between Dev teams and Ops teams, each specialising where needed, but also sharing where needed. There are likely many separate Dev teams, each working on a separate or semi-separate product stack.

My sense is that this Type 1 model needs quite substantial organisational change to establish it, and a good degree of competence higher up in the technical management team. Dev and Ops must have a clearly expressed and demonstrably effective shared goal ('Delivering Reliable, Frequent Changes', or whatever). Ops folks must be comfortable pairing with Devs and get to grips with test-driven coding and Git, and Devs must take operational features seriously and seek out Ops people for input into logging implementations, and so on, all of which needs quite a culture change from the recent past.

<https://web.devopstopologies.com/>



Type 1 suitability: an organisation with strong technical leadership.

Potential effectiveness: **HIGH**

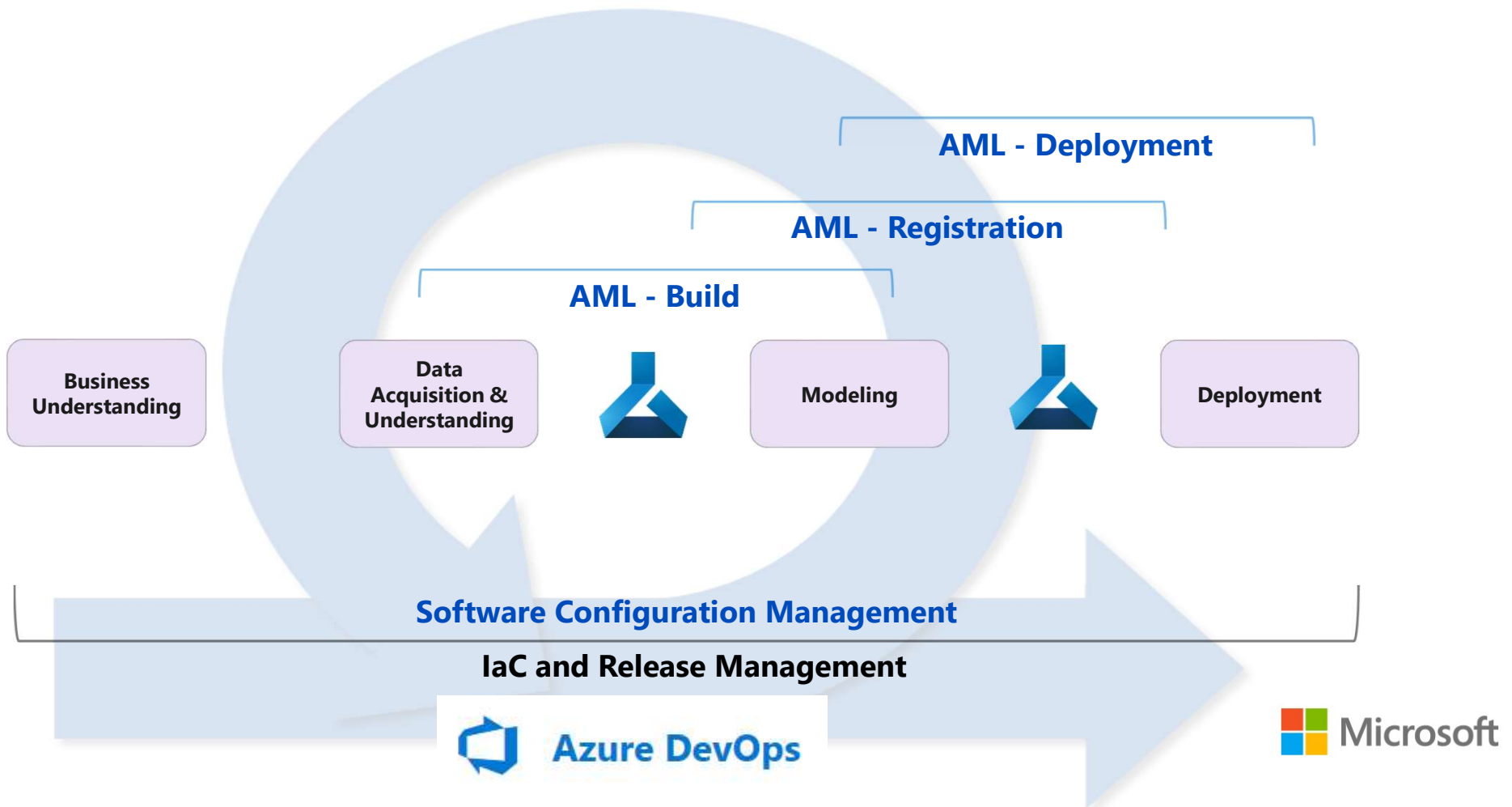
Useful Links

- How to enable DevOps for Data Science Projects – Medium
- <https://medium.com/devopslinks/enable-ci-cd-for-data-science-projects-4f5b61acbf9>
- How to enable DevOps for Data Science Projects – GitHub
- <https://github.com/Azure/DevOps-For-AI-Apps/blob/master/Tutorial.md>
- Getting AI, ML and DevOps working better together – Microsoft Azure
- <https://azure.microsoft.com/en-us/blog/getting-ai-ml-and-devops-working-better-together/>
- DevOps Institute
- <https://devopsinstitute.com/>
- DevOps Working Topologies
- <https://web.devopstopologies.com/>

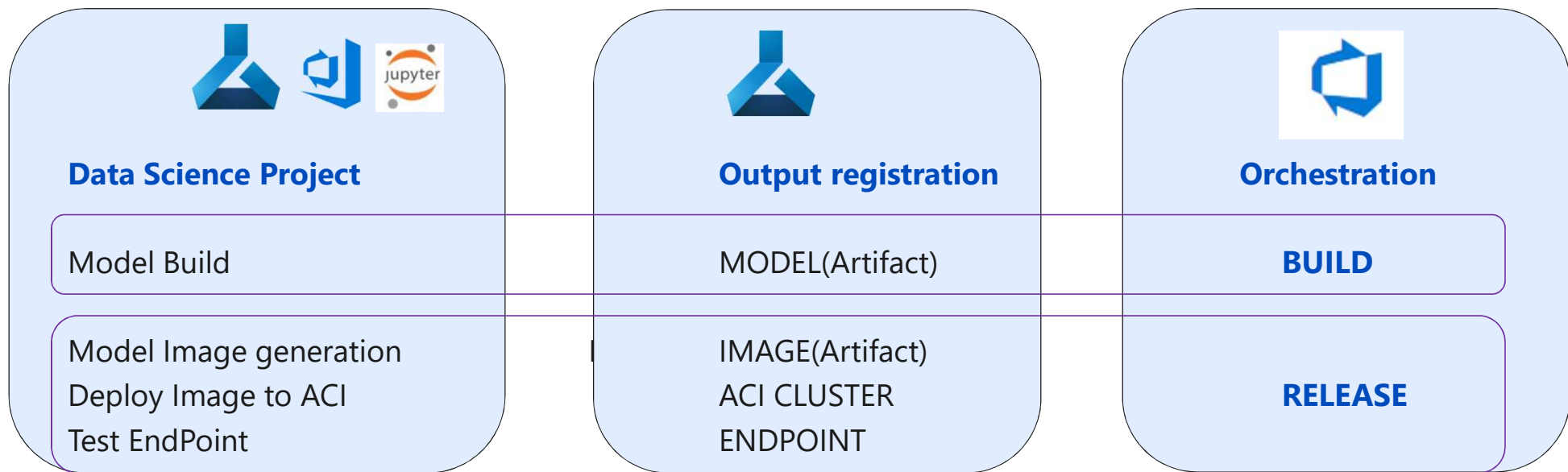
Demo:

How to bring your Azure Machine Learning project to PRODUCTION

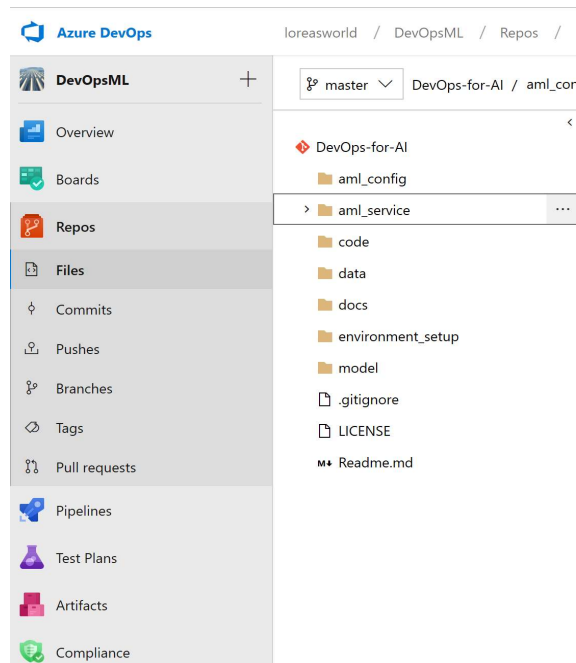
How to bring your Data Science Project to production



How to bring your Data Science Project to production










How to bring your Data Science Project to production



Service
Principle SP

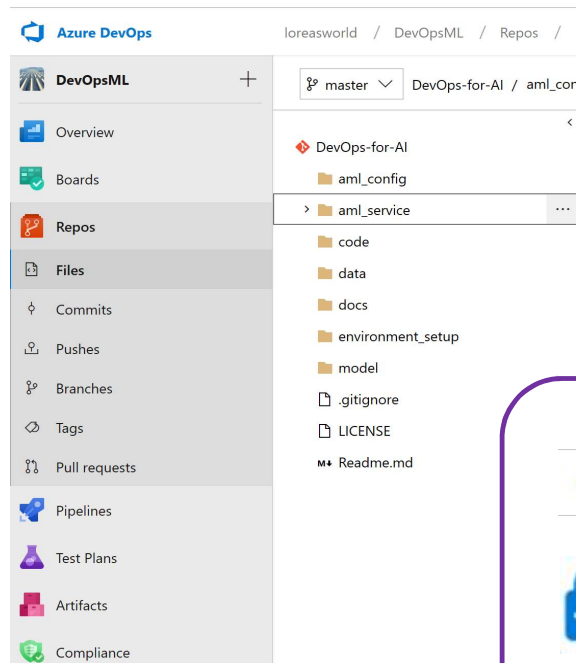

**Azure
ML Service
Model Build**

Application

-  Experiments
-  Pipelines
-  Compute
-  Models
-  Images
-  Deployments
-  Activities


**Azure
ML Service
Model Deploy**

How to bring your Data Science Project to production – with Key Vault



**Azure
ML Service
Model Build**



Key Vault's Keys



**Key Vault's Policies:
Azure DevOps policy**

Application

- Experiments
- Pipelines
- Compute
- Models
- Images**
- Deployments
- Activities



**Azure
ML Service
Model Deploy**



Q&A