MLOps Introduction

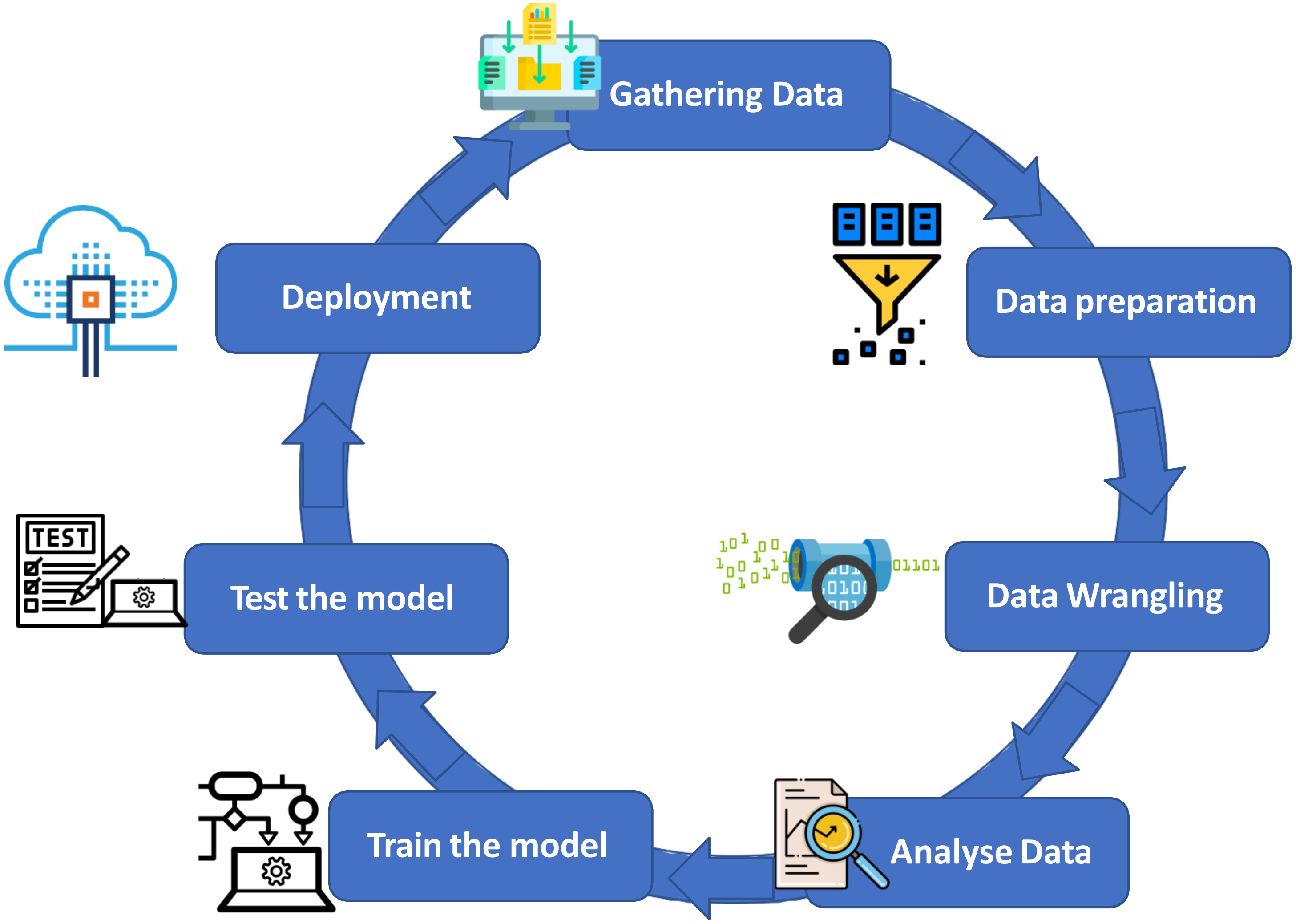
Basudev Panda
basudevpanda@gmail.com

Sachin Shivakalimath
sachin.smath@gmail.com
9449764697

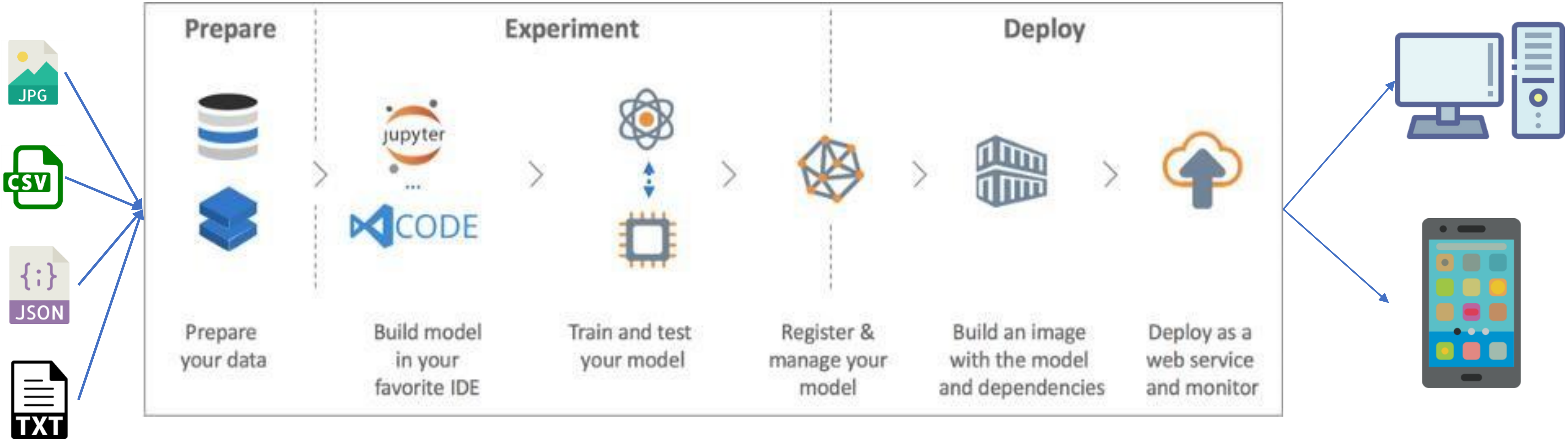
Logistics

1. Git
2. Github account
3. Docker
4. Docker hub account
5. VSCode

Machine Learning Life Cycle

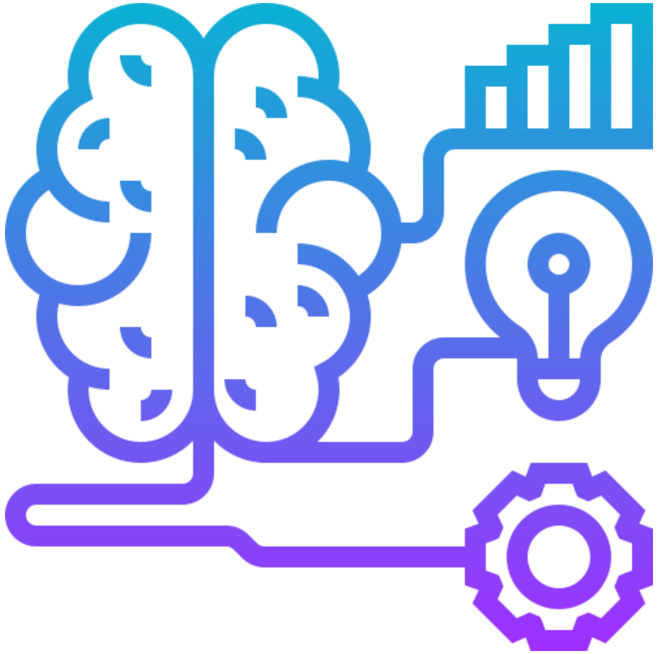


Typical workflow for creating a machine learning model:



Machine learning

Machine learning (ML) is the study of algorithms and mathematical models that computer systems use to progressively improve their performance on a specific task.



Programming Languages

python™

R

Go!

JavaScript

C++

Java™

Machine Learning Operations(MLOps)

Introduction

What is MLOps?

- **MLOps** is a set of practices for collaboration and communication between **data scientists** and **operations professionals**.
- Applying these practices increases the quality, simplifies the management process, and automates the **deployment** of Machine Learning and Deep Learning models in large-scale production environments.

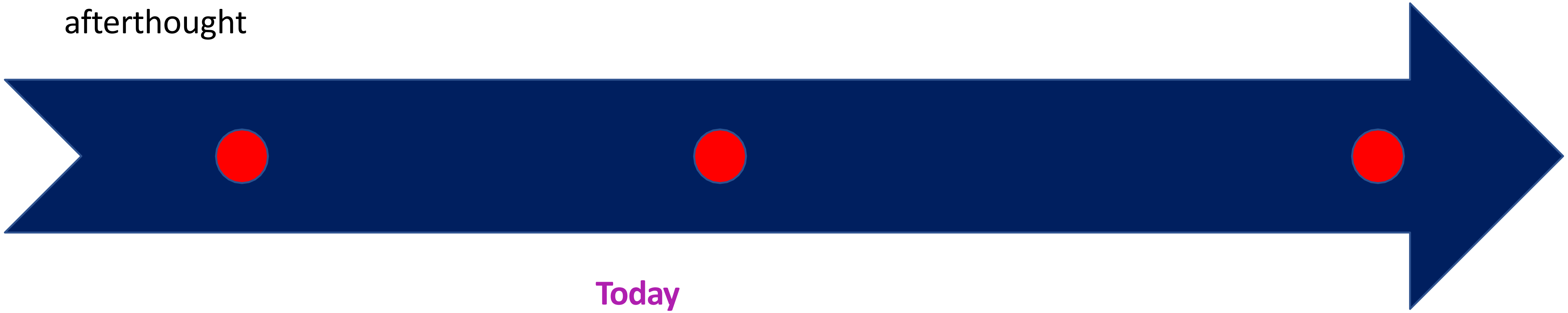
State of machine learning

Last decade

- Focusing mostly on building ML model
- Operationalization was an afterthought

By end of 2024

- **75%** of organizations will shift from piloting to operationalizing AI



Today

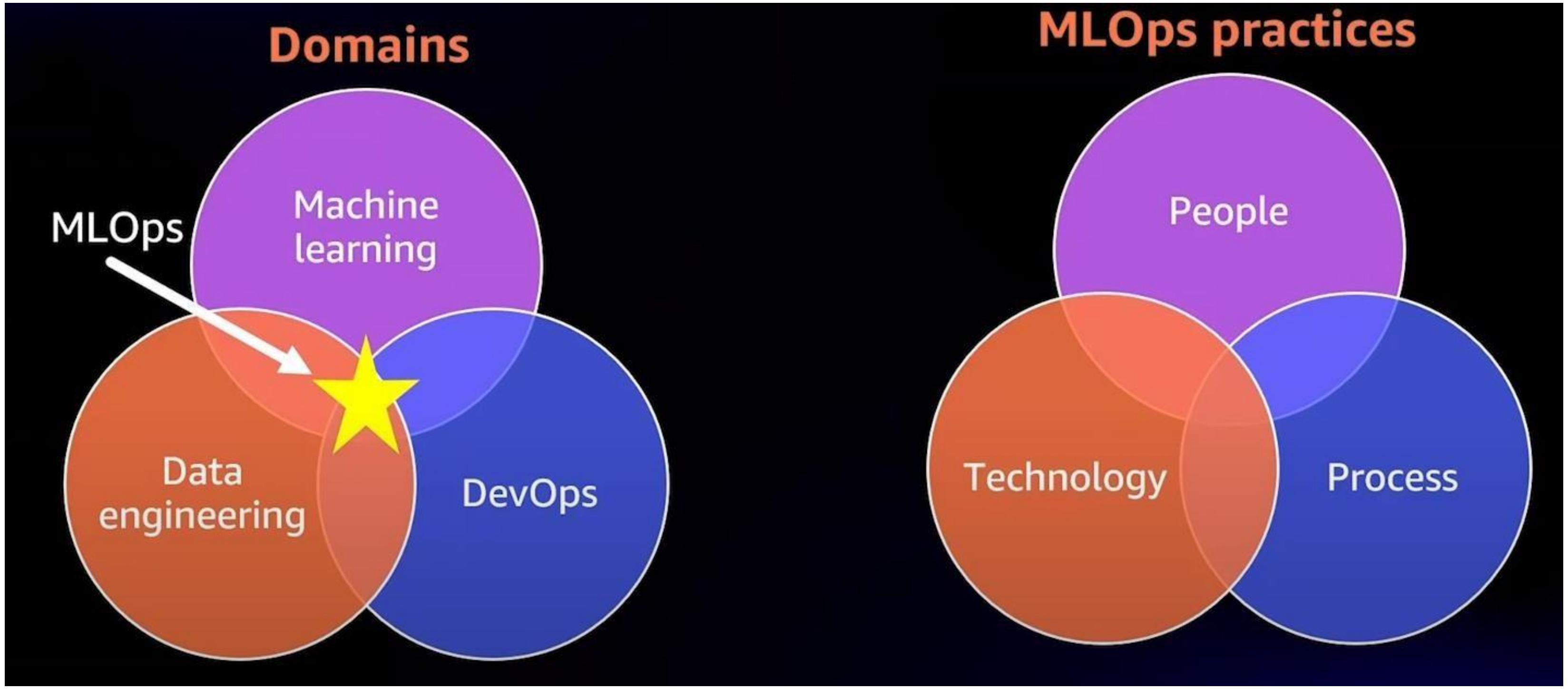
- **53%** of POC make it into production
- Average **9 months**

Source AWS AI conclave 2023, Bangalore, India

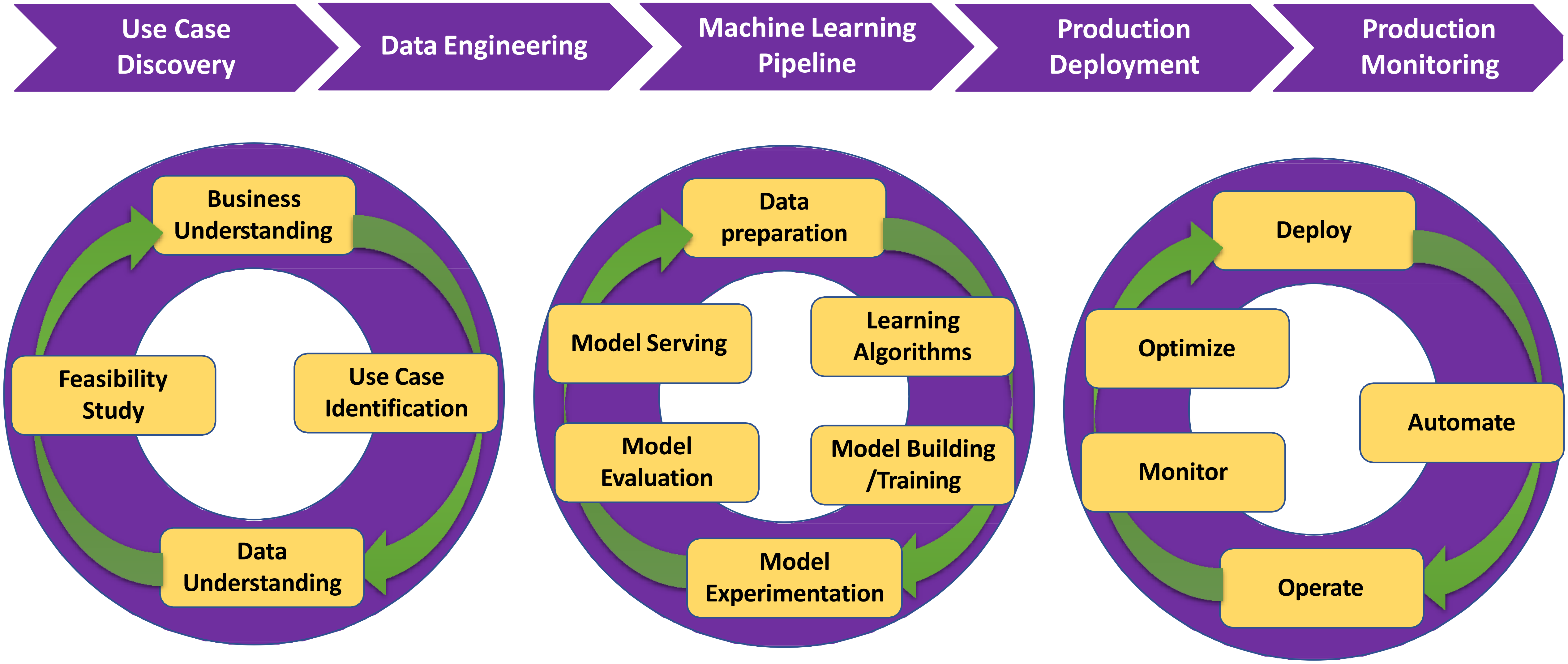
MLOps Motivation: High-level view

MLOps is the discipline that sits at the intersection between the domains of **Machine Learning**, **Data engineering** & **DevOps**.

MLOps as a discipline is enabled by core set of practices that span **people**, **process** & **technology**



MLOps Components



Machine learning industrialization challenges

Data availability and quality:

- ✓ Machine learning models require large amounts of high-quality data

Model deployment and maintenance

- ✓ Deploying models to production ready environment to generate predictions
- ✓ Requires careful planning and ongoing maintenance.

Model performance and accuracy

- ✓ Continuously monitoring the performance and accuracy of deployed machine learning model
- ✓ Challenging when data distribution or underlying business processes are constantly changing

Machine learning industrialization challenges

Explainability and interpretability:

- ✓ Important to understand how a machine learning model is making predictions
- ✓ Many machine learning models are difficult to interpret

Bias and fairness

- ✓ Machine learning models can sometimes reflect the biases present in the data they are trained on

AI Industrialization Challenges

AI Adoption:

- ✓ Poor AI strategy makes startups, enterprises fail today
- ✓ AI adoption with strategy drives the AI into production

AI Strategy

- ✓ Your beginning of the AI model results looks good
- ✓ Then changes or improvements of multiple dimensions
  - Feature additions/attributes
  - Regularization
  - Standardization of models

..may back propagate to the original state

Training and retraining

- ✓ Scale issues will be seen only after moving the models into production

Managing and securing data

- ✓ AI systems often rely on large amounts of data
- ✓ Must be managed and secured in order to protect sensitive information
- ✓ Ensure compliance with relevant regulations

Scaling Issues

- ✓ Running AI at Scale is also one of the major challenges
- ✓ Running model at scale needs
  - Robust orchestration
  - Load balancing
  - Model scale building using deep learning techniques with GPU's

Technology and People

- ✓ **Technology** and **people** with right skills may be able to build the model to run in scale.
- ✓ Challenges after moving model to production
  - Real time data pattern changes
  - Data volume changes
  - Load issues for the model
- ✓ Right skills to address scaling issues
  - Stats
  - Data science skills
  - Analytical skills
  - DevOps skills

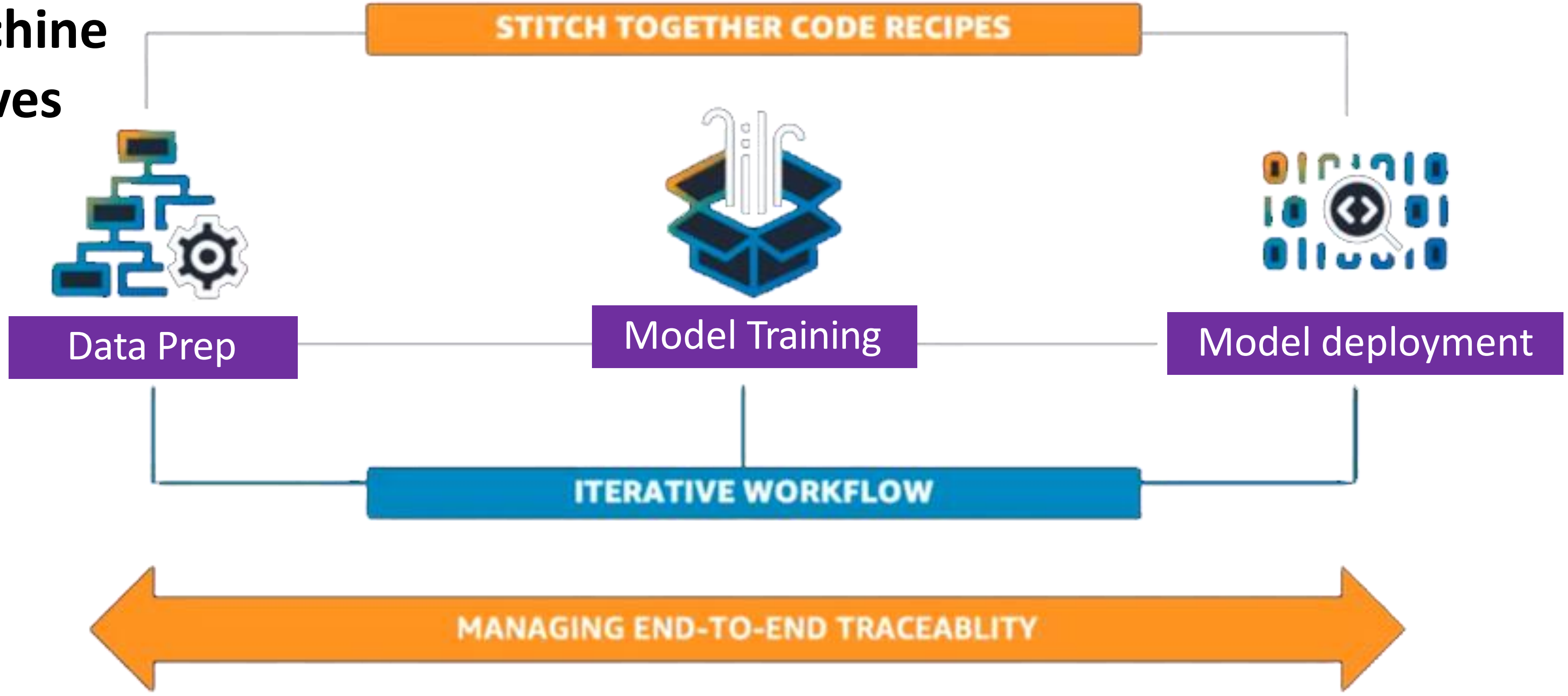
Platforms

- ✓ Platforms for AI models is one of the key challenges now a days enterprises facing
- ✓ Tools in this platform may address the issue the issue of model outages or monitoring model issues etc
- ✓ Consider using ML Pipeline tools and techniques in orchestrating the models

MLOps challenges

Creating & managing Machine Learning workflows involves lot of challenges

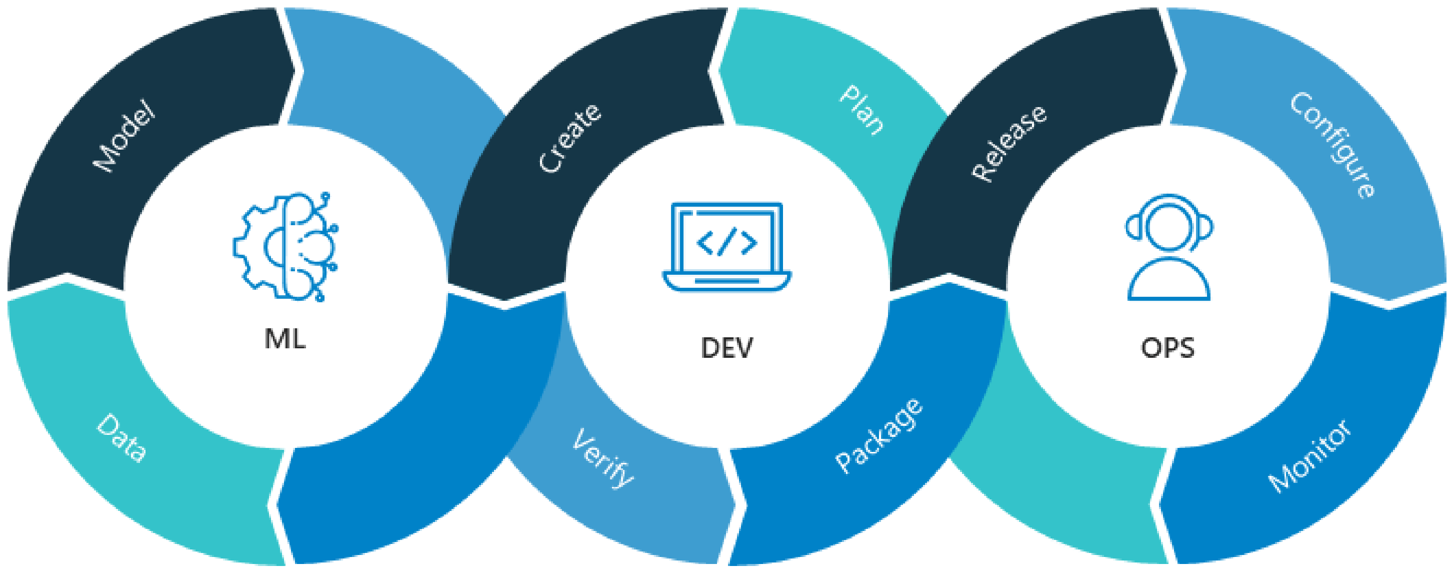
- Data Preparation
- Model Training
- Model Deployment



MLOps challenges similar to DevOps

Challenges in operationalizing ML models have lot common with software production. Incorporating **CI/CD practices** on top of all of this like **Source & version control.**

- **Source code versioning**
- **Track & versioning data**
- **Model artifacts versioning**
- **Model versioning**



Use case discovery:

- Collaboration between **business** and **data scientists** to define a business problem and translate that into a **problem statement**
- Objectives solvable by **ML** with associated relevant KPIs (Key Performance Indicator).

Data Engineering:

- Collaboration between **data engineers** and **data scientists** to acquire data from various sources
- Prepare the data (processing/validation) for modeling.

Machine Learning pipeline:

- This stage is **designing** and **deploying** a pipeline **integrated with CI/CD**.
- Data scientists use pipelines for multiple experimentation and testing.
- The platform keeps track of **data** and **model lineage** and **associated KPIs** across the experiments.

Production deployment:

This stage accounts for **secure** and **seamless** deployment into a production server of choice, be it **public cloud**, **on-premise, or hybrid.**

Production monitoring:

- This stage includes both **model** and **infrastructure monitoring**.
- Models are continuously monitored using configured **KPIs** like changes in **input data distribution** or **changes in model performance.**

How does it relate to **DevOps**, **AIOps**, **ModelOps**, and **GitOps**?

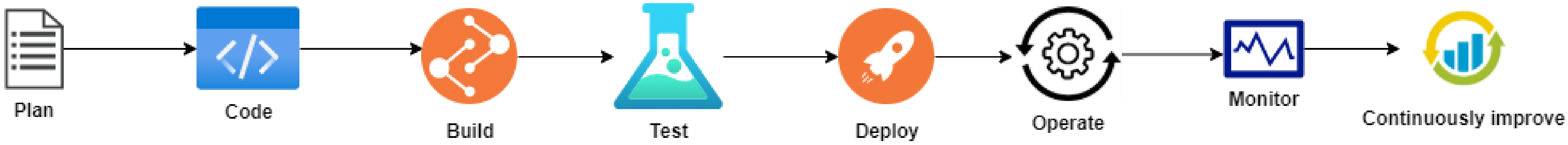
How does it relate to **DevOps**, **AIOps**, **ModelOps**, and **GitOps**?

DevOps

DevOps is a set of practices and principles that emphasizes collaboration and automation to improve the speed, reliability, and security of software delivery.

Life cycle

Involves multiple phases, from **development** and **testing** to **deployment** and **monitoring**



Tools & Platforms

- Configuration Management Tools: **Ansible, Chef, and Puppet**
- Containerization and Orchestration Tools: **Docker and Kubernetes**
- Continuous Integration and Continuous Deployment (CI/CD) Tools: **Jenkins, Travis CI, CircleCI, and GitLab CI/CD**
- Monitoring and Logging Tools: **Prometheus, Grafana, and Elasticsearch/Kibana**
- Collaboration and Communication Tools: **Slack, Microsoft Teams and JIRA**
- AIOps Tools: **Moogsoft and Big Panda**

DevOps Aim

- DevOps aims to improve the speed and reliability of software delivery, and to increase collaboration and communication between development and operations teams

  - ✓ Deliver software faster
  - ✓ Increase reliability and reduce errors
  - ✓ Improve collaboration and communication
  - ✓ Achieve better visibility and control
  - ✓ Enhance Security
  - ✓ Achieve better scalability

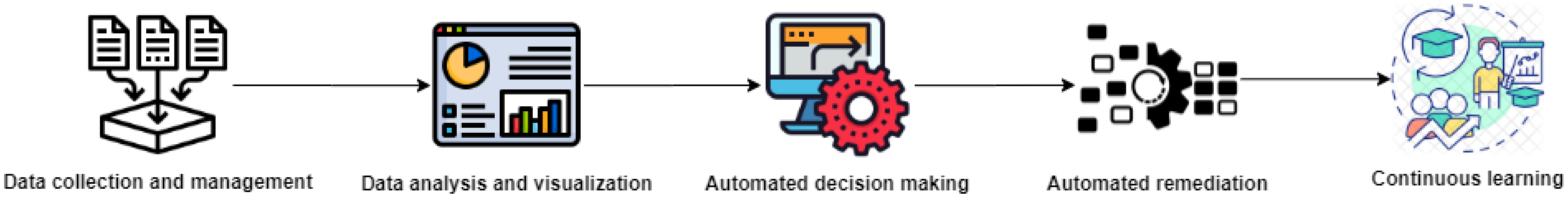
AIOps

AIOps (Artificial Intelligence for IT Operations) is the use of machine learning and other AI technologies to automate many processes that are currently done manually in an organization.

AIOps is also different from MLOps because it uses AI to automate many processes, not just one or two tasks like MLOps does.

The organizations planning to implement AIOps will need to have in place MLOps

Life cycle



Tools & Platforms

- ✓ ELK Stack
- ✓ Prometheus
- ✓ Grafana
- ✓ Splunk
- ✓ AppDynamics
- ✓ IBM Netcool Operations Insight
- ✓ ServiceNow AIOps

AIOps Aim

- The main aim of AIOps is to improve the **efficiency** and **effectiveness** of IT operations by using artificial intelligence and machine learning to automate and optimize various tasks.

  ✓ Improved incident management
  ✓ Increased automation
  ✓ Reduced mean time to repair (MTTR)
  ✓ Increased visibility
  ✓ Improved Root Cause Analysis
  ✓ Predictive maintenance

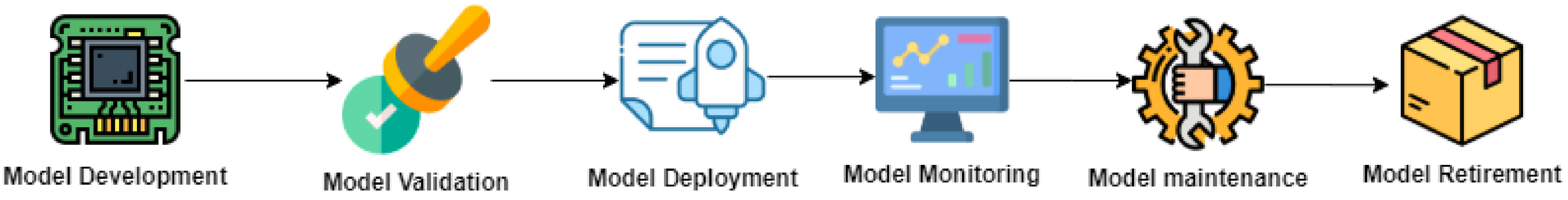
ModelOps

- ModelOps (Model Operations) is the practice of **managing** and **deploying machine learning models** in a **production environment.**
- It includes the processes, tools, and best practices
  - ✓ Developing
  - ✓ Testing
  - ✓ Deploying
  - ✓ Monitoring
  - ✓ Maintaining machine learning models in a production environment.

key components of ModelOps

- ✓ Model development
- ✓ Model deployment
- ✓ Model monitoring
- ✓ Model maintenance
- ✓ Model Governance

Life cycle



Tools & Platforms

- ✓ TensorFlow Extended (TFX)
- ✓ MLflow
- ✓ Amazon SageMaker
- ✓ DataRobot
- ✓ Algoworks AI-ML Platform
- ✓ Google Cloud AI Platform
- ✓ RapidMiner

ModelOps Aim

- The goal of ModelOps is to ensure that machine learning models are **reliable**, **accurate**, and **maintainable**, so that they can be used effectively in real-world applications.

GitOps

- ✓ GitOps is a practice that uses Git as a **single source of truth** for declaratively **managing infrastructure** and **application deployments.**

- ✓ It uses Git as a **central repository** to **store configuration files** and use them to drive the state of the systems

- ✓ It's based on the idea that the version-controlled configuration files that describe the desired state of the systems should be treated as the single source of truth

key components of GitOps

- ✓ Centralized repository
- ✓ Declarative configuration
- ✓ Automated deployment
- ✓ Continuous integration and delivery (CI/CD) pipelines
- ✓ Continuous monitoring
- ✓ Collaboration

Tools & Platforms

- ✓ Code is committed to a version control repository
- ✓ CI/CD system listens for changes to the repository and automatically builds and tests the code
- ✓ If test passed the code is deployed to a staging environment for further testing
- ✓ Once the code is deemed ready, it is deployed to the production environment
- ✓ Any updates or rollbacks to the production environment are done by committing changes to the version control repository.
  - Triggers the CI/CD pipeline to redeploy the changes.

GitOps Aim

- ✓ GitOps uses Git as a **single source of truth** for declaratively managing distributed systems, such as Kubernetes clusters
- ✓ All desired states of the system are stored in Git
- ✓ Changes to the system are made by committing **new versions** of these manifests to the Git repository
- ✓ GitOps allows for **version control**, **review**, and **rollback** of infrastructure changes, and also allows for easy collaboration and auditing
- ✓ Allows you to use tools you are already familiar with, like **git** and **CI/CD pipelines**.

Major Phases - what it takes to master MLOps

Mastering MLOps (Machine Learning Operations) can be a **complex** and **multifaceted** endeavor, as it involves both machine learning expertise and operations experience.

key areas that are important to master in MLOps:

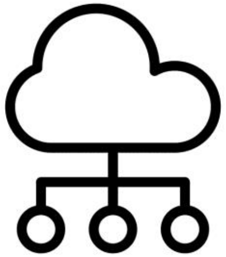
- ✓ Machine Learning
- ✓ Cloud Computing and Infrastructure
- ✓ CI/CD and Automation
- ✓ Monitoring and Evaluation
- ✓ Security and Compliance
- ✓ Data Engineering and Operations

Major Phases - what it takes to master MLOps

Machine Learning

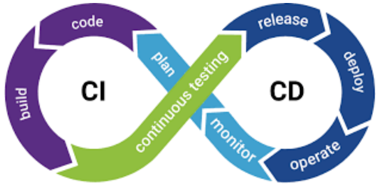
- ✓ To master MLOps, you should have a solid understanding of machine learning concepts, algorithms, and frameworks.
- ✓ This will allow you to develop, train, and deploy machine learning models effectively.

Cloud Computing and Infrastructure



- ✓ Experience with cloud computing platforms, such as AWS, Azure, or GCP.
- ✓ Experience with container orchestration tools like Kubernetes.
- ✓ These are crucial for MLOps as they are often used to deploy and manage machine learning models in production.

CI/CD and Automation



- ✓ To be effective in MLOps, you should be proficient in using continuous integration and continuous delivery (CI/CD) tools

- ✓ Automation frameworks to build, test, and deploy machine learning models and infrastructure.

Major Phases - what it takes to master MLOps

Monitoring and Evaluation

- ✓ MLOps also require being able to **monitor** the **performance** and **behavior** of **machine learning models** in production
- ✓ Including **logging**, **alerting** and **tracking** of the models performance over time to measure their **effectiveness**.

Security and Compliance

- ✓ Machine learning models can process **sensitive data**
- ✓ Important to have a solid understanding **of data security** and **compliance best practices**
- ✓ To ensure that the **models** are **deployed** and **managed securely.**

Major Phases - what it takes to master MLOps

Data Engineering and Operations

- ✓ Considering data is the **lifeblood of machine learning models**
- ✓ A solid understanding of **data engineering** and **operations** is crucial in MLOps
- ✓ Strong **data pipeline** and **storage skills** will be very beneficial in this regard

CI/CD in Production Case Study

CI/CD in Production Case Study

One of the core concepts in DevOps that is now making its way to machine learning operations (MLOps) **is → CI/CD—Continuous Integration and Continuous Delivery or Continuous Deployment.**

- ✓ **Continuous integration (CI)** is the practice of automating the **building** and **testing** of code every time it is committed with **version control and pushed to a code repository** (to build the application).
- ✓ **Continuous delivery (CD)** is the practice of **deploying every build to a production-like environment** and **performing automated integration** and **testing** of the application before it is deployed.
- ✓ **Continuous deployment (CD)** compliments continuous integration with additional steps by **automating the configuration** and **deployment** of the application to a production environment.

CI/CD for Machine Learning (ML) with Azure DevOps

Industry: Retail and consumer goods.

Use case:

- ✓ This team helps a **retail client** to **resolve tickets** in an automated way using machine learning
- ✓ This tickets are raised by users or raised by maintenance problems
- ✓ Machine learning is used to classify the tickets into different categories, helping in the faster resolution of the tickets.

Core CI/CD tools:

- ✓ Azure DevOps Pipeline
- ✓ Git

Overview:

- ✓ To orchestrate their CI/CD workflow, the team used the **Azure DevOps suite of products.**
- ✓ They also configured **development** and **production** environments for their ML workloads.
- ✓ These workflows consist of all CI/CD processes that happen **before deploying the model to production** and **after deployment.**

CI/CD workflow before deploying the model to production

Step 1:

- To automate the **dev-to-production cycle**, the team set up **build and release tasks** with **Azure DevOps Pipelines.**

- The build pipeline generates the model artifacts from a candidate source code and after model serialization (mostly using ONNX).

Step 2:

- The artifacts are deployed to infrastructure targets using **the release pipelines**

- The release pipelines move the artifacts to the **quality assurance (or QA**) stage after they have been tested in the **development environment.**

Step 3:

- **Model testing happens in the QA stage** where **A/B** tests and **stress tests** are performed on the model

- To ensure model is ready to be deployed to the production environment.

Step 4:

- A human validator, usually the product owner, ensures the model passes the tests, has been validated

- And then approves the model to be deployed to the production environment using the release pipelines.

CI/CD workflow after deploying the model to production

Step 1:

- After deploying the model to production
- Team sets up **cron jobs** that **monitor model metrics** for **data drift** and **concept drift** on a weekly basis
- So that the pipeline can be triggered when an unacceptable drift occurs that requires retraining the model.

Step 2:

- They also monitor the performance of their CI/CD pipeline in production
- The purpose of the inspection is to ensure their CI/CD pipeline is healthy and in a robust state

CI/CD for ML with GitOps using Jenkins and Argo workflows

Industry

Computer software

Use case

- **GreenSteam** – An i4 Insight Company provides **software solutions for the marine industry that help reduce fuel usage**
- Excess fuel usage is both costly and bad for the environment, and vessel operators are obliged to get more green by the International Maritime Organization and reduce the CO_2 emissions by 50 percent by 2050

Core CI/CD tools

✓ Argo
✓ Jenkins

Overview:

- The team used **Jenkins** and **GitOps** to automatically check and test their code before deploying it to production, by simulating production-like conditions in a test environment.

- The team had a **single pipeline for model code** where every **pull request** was going through **code reviews** and **automated unit tests.**

- Pull requests were checked using **automated tests** that **trained models**, **made predictions**, and **ran the whole process on small pieces of real data** to ensure everything worked correctly and nothing was broken

- **Models were continuously delivered** and **reviewed** by a **domain expert** after training, and were deployed manually after getting approved by the expert and passing all checks.

Code quality checks and using Jenkins to manage the CI pipeline

- **Jenkins** is one of the most popular tools used for continuous integration among developers for **tests**, **checks**, and **reviews**

Step: 1

✓ They put all code quality checks in a Docker container to keep consistency and unify the tools and configurations used locally and on Jenkins.

✓ So versioning & config tools required like flake8, black, mypy, pytest all are unified

Step: 2

✓ Docker eliminated issues caused by different versions of dependencies on local, Jenkins, and production environments.

Step: 3

- ✓ For local development, they had a Makefile to build the Docker image and run all the checks and tests on the code.

Step: 4

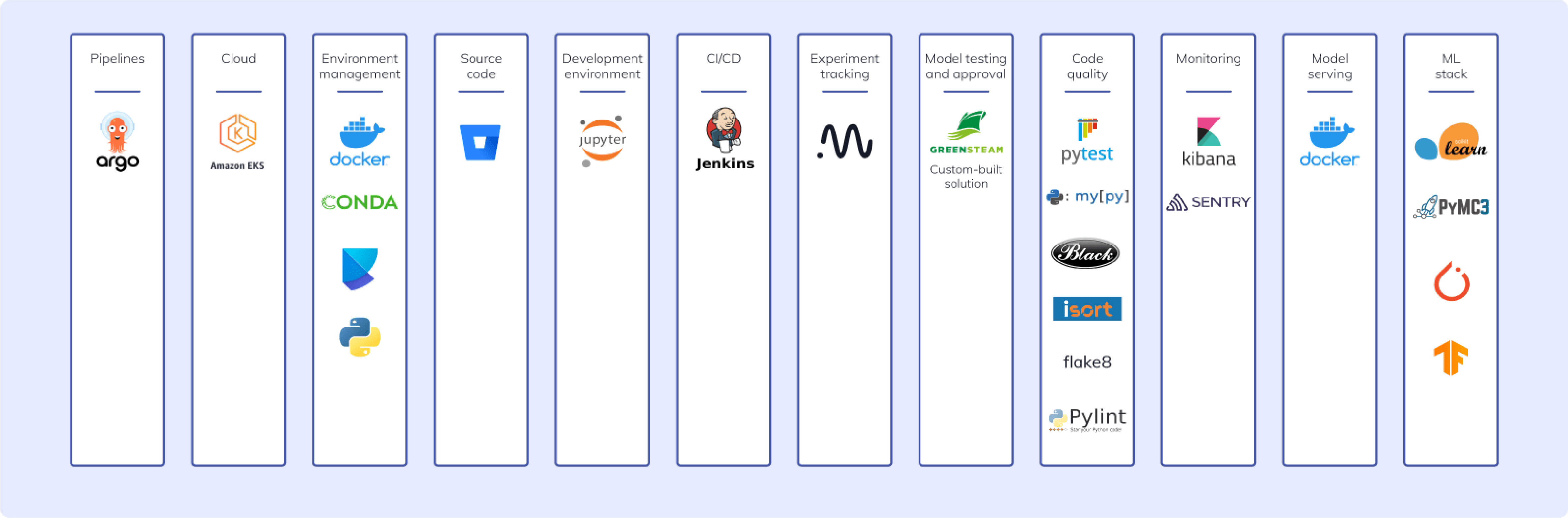
- ✓ For code reviews, they set up Jenkins and it was running the same checks as a part of the CI pipeline.

Using Argo to manage CI/CD pipelines

- ✓ Argo Workflows is a tool that runs parallel jobs on Kubernetes
- ✓ Allows the team to run heavy computational jobs like machine learning and data processing on Amazon EKS clusters.
- ✓ The pipeline includes:
    - Retraining models
    - Testing
    - Expert review before deployment

Team’s entire stack for ML workloads:

GREENSTEAM MLOps Toolstack



CI/CD for ML with AWS CodePipeline and Step Functions

Industry

Transportation and logistics.

Use case

For this use case, the team is from a consulting and professional services company that worked on a public project. Specifically, they built machine learning applications that solved problems like:

- ✓ Predicting how long it will take to deliver a parcel
- ✓ Predicting a location, based on unstructured address data and resolving it to a coordinate system (latitude/longitude).

Core CI/CD tools

- ✓ **AWS CodeBuild** – A fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are **ready to deploy**.
- ✓ **AWS CodePipeline** – A fully managed continuous delivery service that helps you automate your release pipelines.
- ✓ **AWS Step Functions** – A serverless function orchestrator that makes it easy to sequence AWS Lambda functions and multiple AWS services.

Overview:

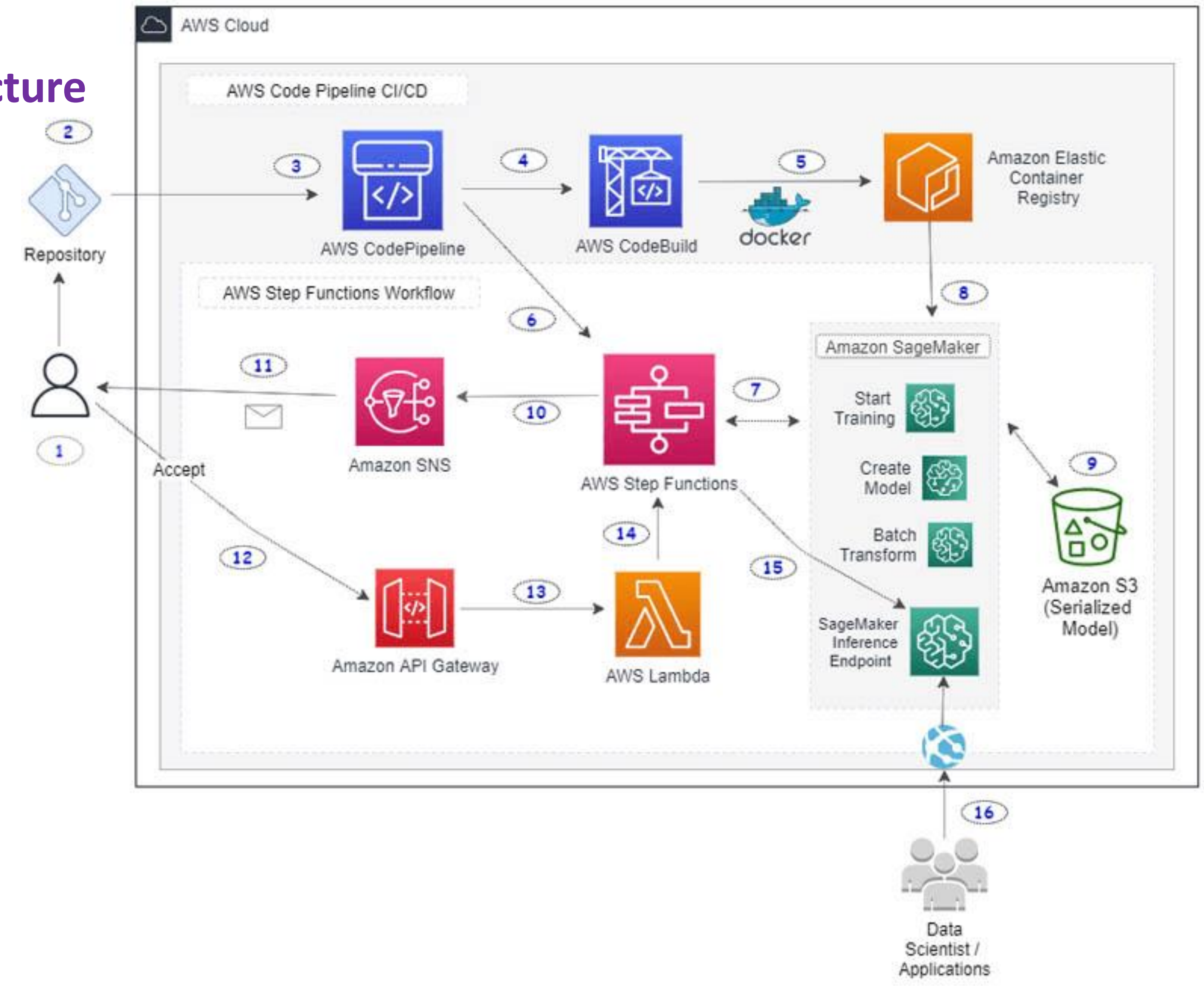
- AWS Cloud provides managed CI/CD workflow tools like **AWS CodePipeline** and **AWS Step Functions** to carry out continuous integration and continuous delivery for their **machine learning projects.**

- For **continuous integration**, the team used **git** to make commits to **AWS CodeCommit**

- Commits trigger a build step in **CodePipeline** (through an AWS CodeBuild job)

- AWS **Step Functions** handle the **orchestration** of the workflows for every action from CodePipeline.

Understanding the architecture



CI/CD for ML with Vertex AI and TFX on Google Cloud

Industry

- ✓ Business intelligence and financial technology services.

Use case

Digits Financial, Inc. is a fin-tech company offering a visual, machine learning-powered expense monitoring dashboard for startups and small businesses.

- ✓ Building a tool to turn a company's financial data into a real-time model.
- ✓ Using unstructured data to predict future events for customers.
- ✓ Grouping data to show what's most relevant for customers' businesses.

Core CI/CD tools

- ✓ TensorFlow Extended
- ✓ Vertex AI Pipelines

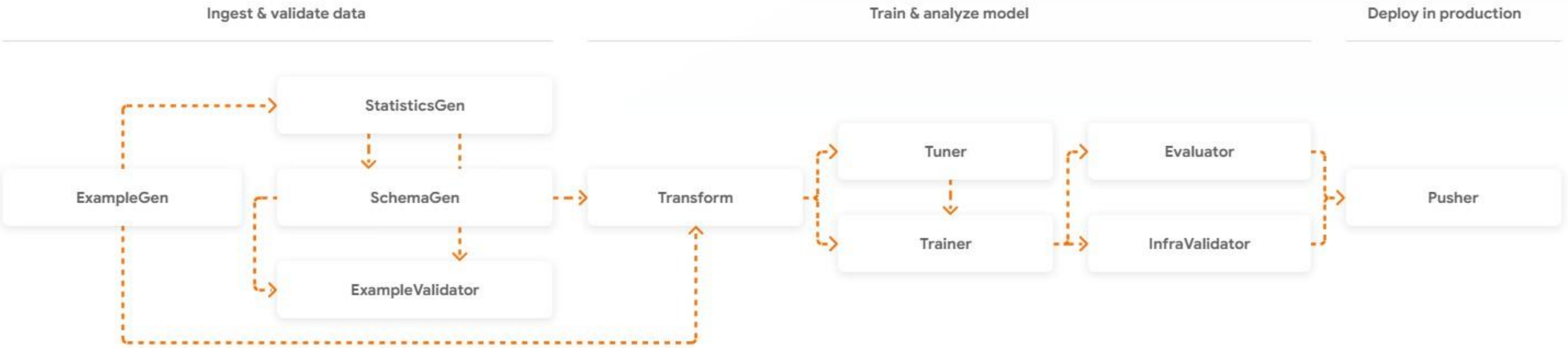
TensorFlow Extended

Vertex AI

Overview

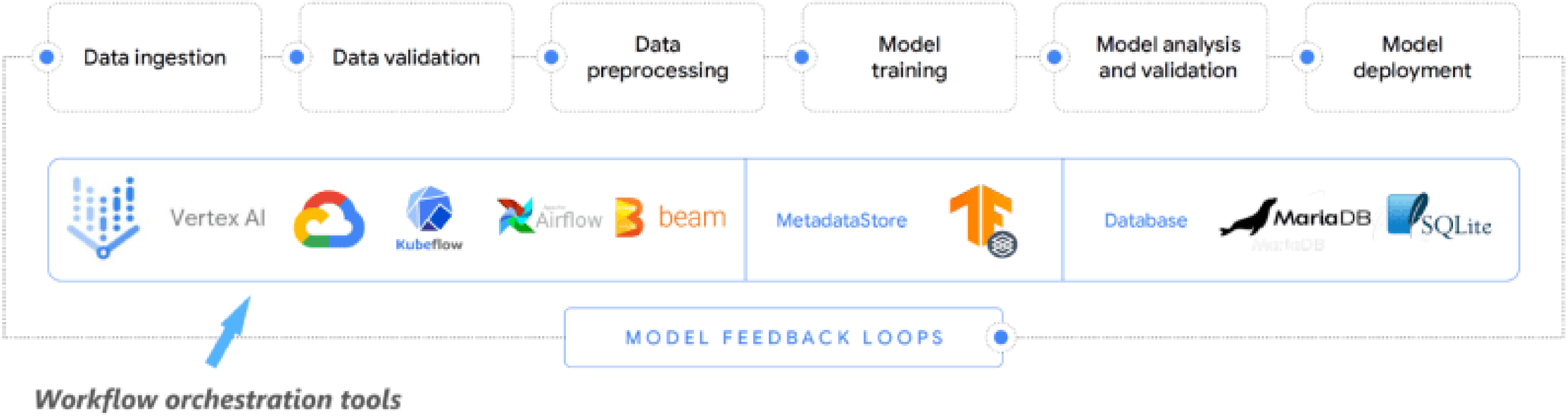
- ✓ The team at Digits used a specialized tool called **Vertex AI Pipelines** and **TensorFlow Extended**
- ✓ They used this tool to **manage** and **improve** the process of implementing **machine learning models on Google Cloud**
- ✓ This approach helped to maintain **consistency** and **quality** in the models created.

What is TensorFlow Extended (TFX) ?



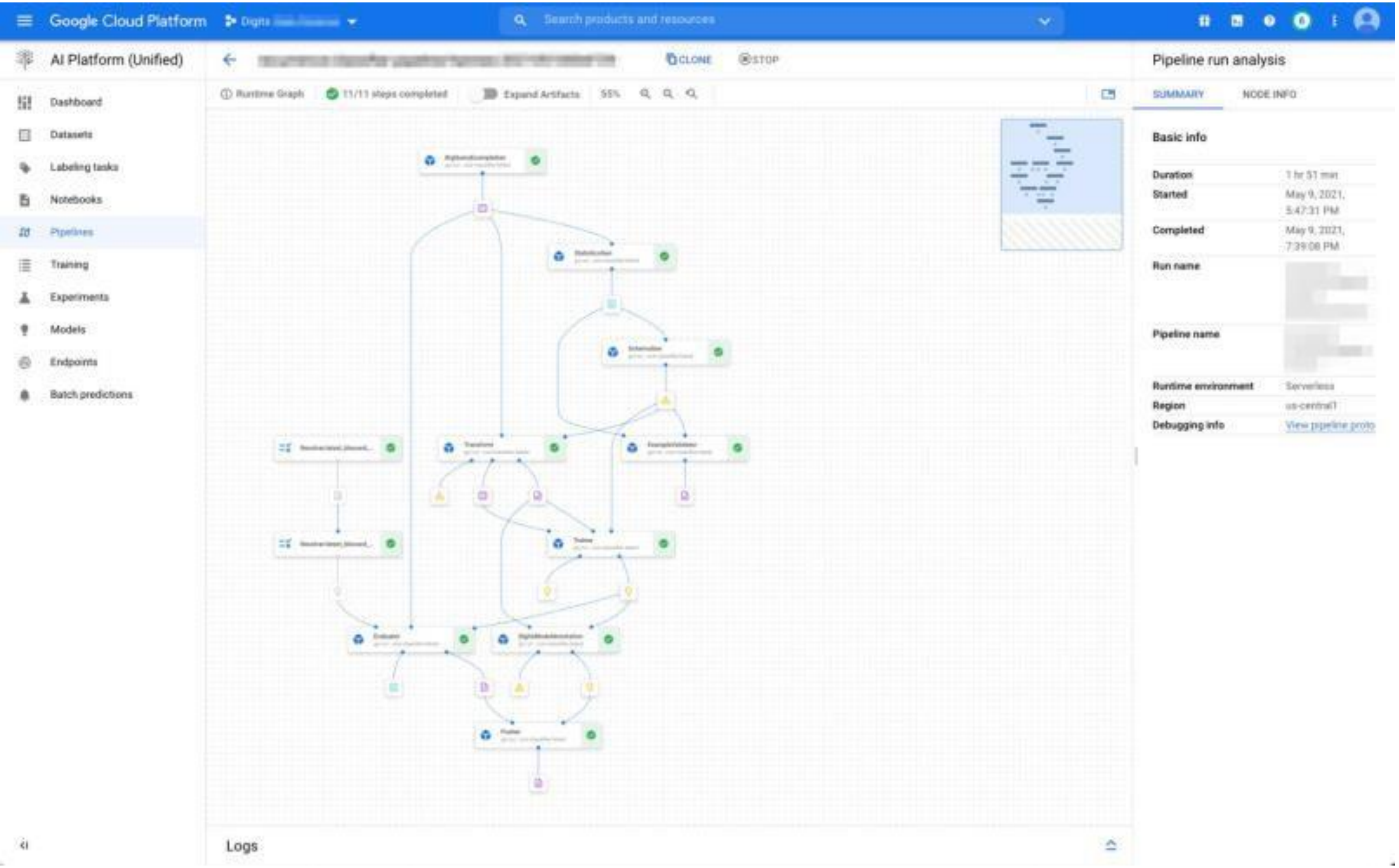
- ✓ It's a **library** for creating and deploying production machine learning pipelines
- ✓ Provides a set of tools for building, deploying, and managing machine learning workflows
- ✓ TFX pipeline is a set of reusable components
- ✓ These components include **data ingestion** and **validation**, **model training**, **evaluation**, and **serving**
- ✓ As well as **model monitoring** and **management**.
- ✓ Pipelines can be run using **Apache Beam, Apache Airflow, or Kubeflow Pipelines**
- ✓ Deployed to a variety of environments, including on-premises and cloud-based systems.

Machine learning pipelines with TFX



- ✓ The team moved their **ML pipelines** from **Kubeflow** to **Vertex AI Pipeline** from Google Cloud
- ✓ This helped them easily tie together **model development (ML)** and **operations (Ops)** into high-performance and reproducible steps
- ✓ One of the core advantages of using Vertex AI Pipelines is that it helped the **team transition** from **managing their pipeline** to leveraging the **managed Vertex AI Pipeline service** for workflow orchestration
- ✓ This removed the need to maintain databases **that store metadata, launch clusters to host and operate the build servers and pipelines**
- ✓ Vertex AI Pipeline service is **a managed service**, which reduces the maintenance required to self-hosted Kubeflow Pipelines.

Orchestrating with Vertex AI Pipelines



- ✓ Vertex AI is a platform that helps to speed up **experimentation** and **deployment** of machine learning models
- ✓ It helps to **automate**, **monitor**, and **govern the team's ML systems** by organizing them in a **serverless manner**
- ✓ It stores the workflow's artifacts
- ✓ By storing these artifacts, the team can trace the origin of the models, including the training data, hyperparameters and code used to create the model.

Benefits from using machine learning pipelines

- ✓ Using ML pipelines reduced the DevOps requirements for the team
- ✓ Migrating to managed ML pipelines reduced the expense of running 24/7 clusters
- ✓ Model updates were easy to integrate and automated, freeing up the team for other projects
- ✓ Consistency across all ML projects because the teams could run the same tests and reuse the pipeline or components
- ✓ A centralized place for machine learning-related metadata and information
- ✓ Models are automatically tracked and auditable.