Fluid In ML Model Serving

Lize Cai
Sept, 2022

Disclaimer

This is not a presentation by SAP. It is my personal account of our product and experience at SAP, some of its implications and learnings.

Content

- What Is SAP AI Core
- Challenges In ML Model Serving
- Fluid on Model Artifact Caching
- Result and Feedback

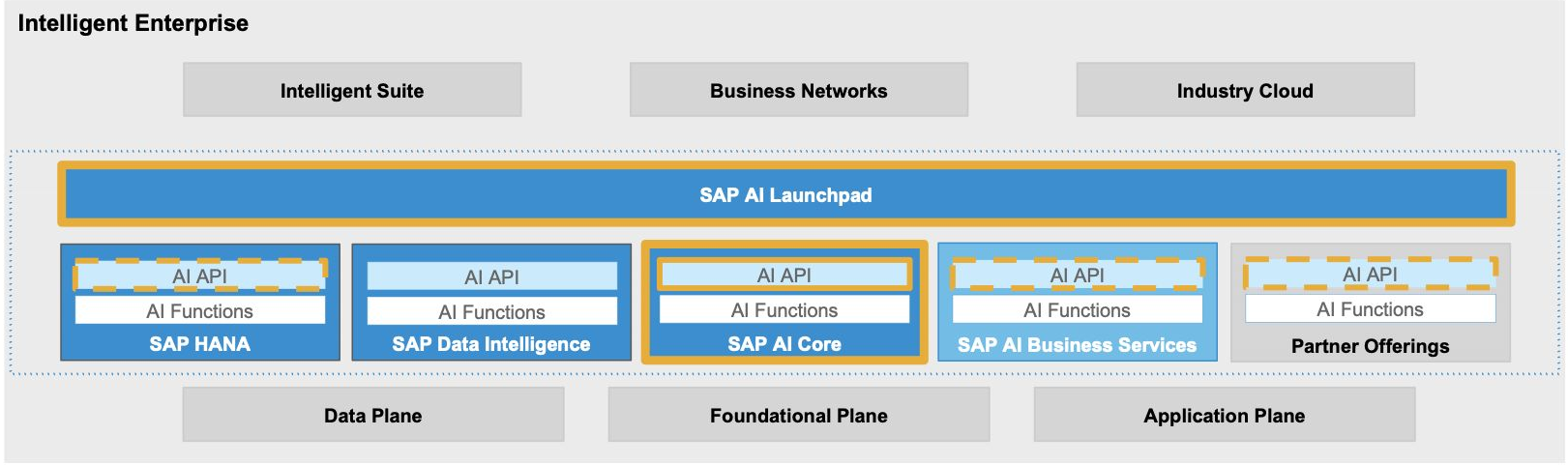
What Is SAP AI Core

SAP AI Core is a service in the SAP Business Technology Platform which is designed to handle the execution and operations of your AI assets in a standardized, scalable, and hyperscaler-agnostic way. It provides seamless integration with your SAP solutions. Any AI function can be easily realized using open-source frameworks. SAP AI Core supports full lifecycle management of AI scenarios.

SAP AI Core & SAP AI Launchpad Technical Building Blocks

Providing Unified AI Consumption and Lifecycle Management for SAP

- AI Functions can be realized as AI Services (e.g. Document Information Extraction), via Packages (e.g. Generic Line Item Matching), or Code using frameworks of choice (e.g. TensorFlow, R).
- Consumption of all AI Functions is unified by an SAP-governed AI API, regardless whether they are deployed on SAP (SAP HANA, SAP Data Intelligence, SAP AI Core) or partner technology (e.g. Azure, GCP, AWS).
- Management and operations of AI Functions (versioning, deployments, monitoring) will be unified across SAP via SAP AI Launchpad.

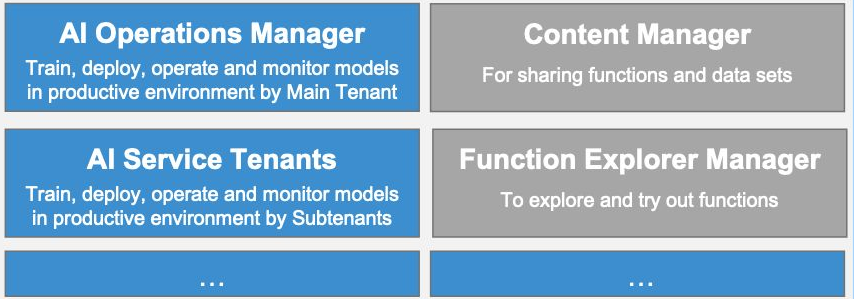


PoC's

AI Core & AI Launchpad Overview

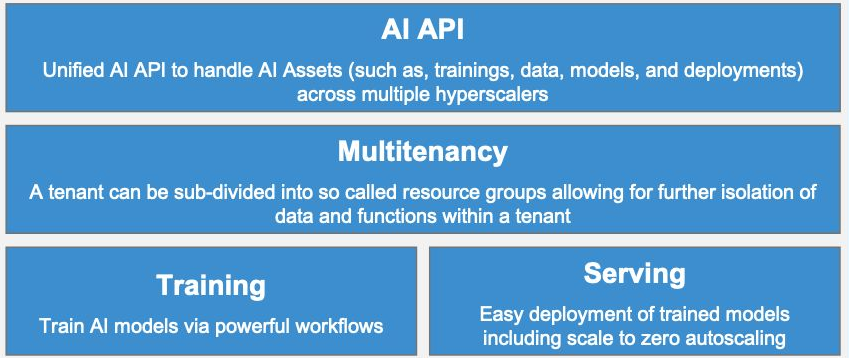
SAP AI Launchpad

SAP AI Launchpad is the central vehicle for SAP teams as well as customers and partners to manage their AI Functions across all landscapes and deployment options.



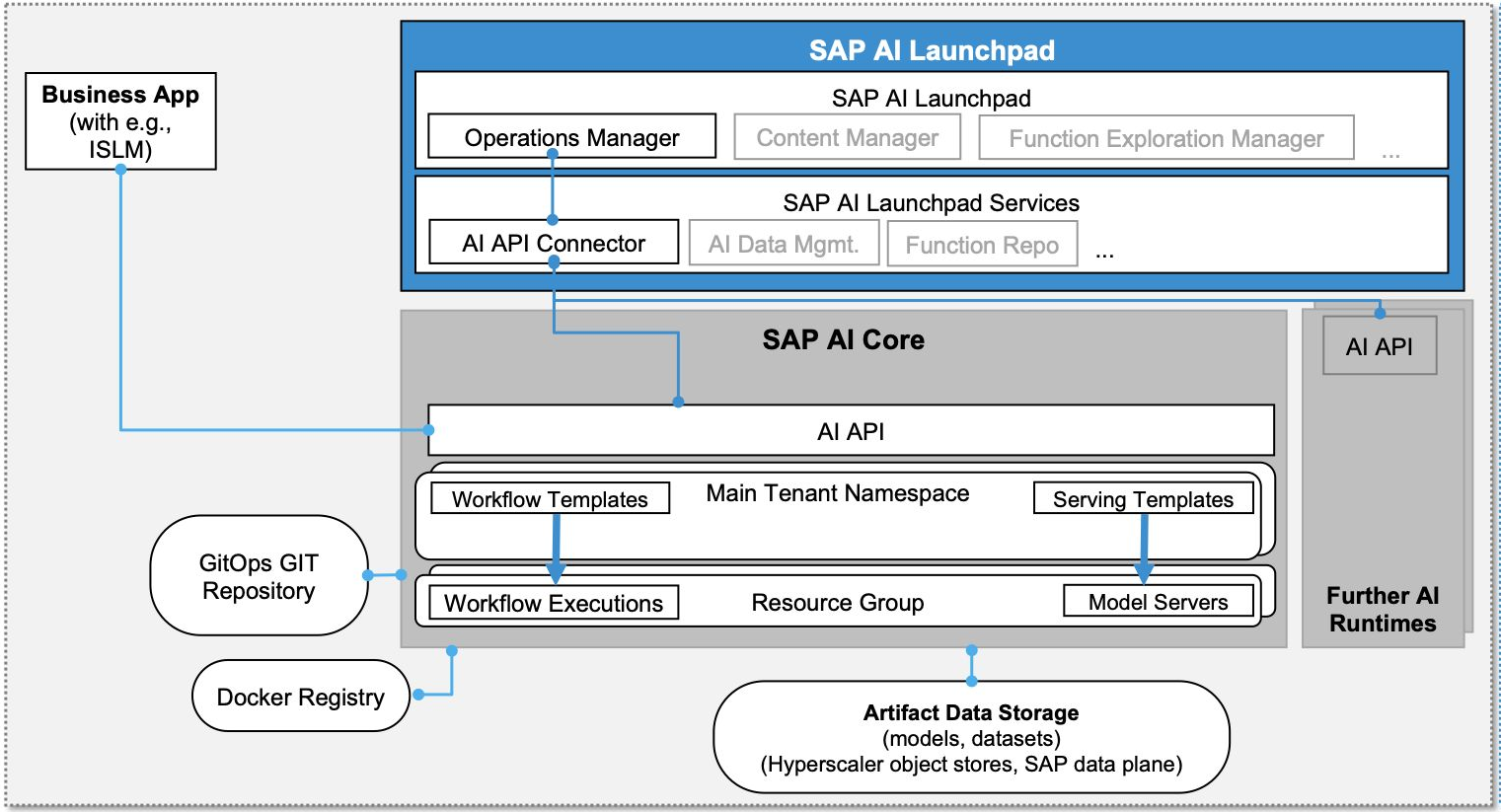
SAP AI Core

- Is accelerating the development and productization of compliant AI Functions:
  - Out-of-the-box integration into SAP solutions
  - Can be developed using any open source ML framework
- Provides full lifecycle management support such as deployment via the GitOps principle
- Built around state-of-the-art open source solutions (e.g. Argo Workflows; KFServing)



Available
Planned

SAP AI Launchpad and SAP AI Core – High Level Architecture Overview



- **SAP AI Launchpad** = management & service layer on top of multiple AI API servers
- **SAP AI Core** = AI API
  - + subtenants
  - + pipelines
  - + serverless
  - + GitOps
  - + ...

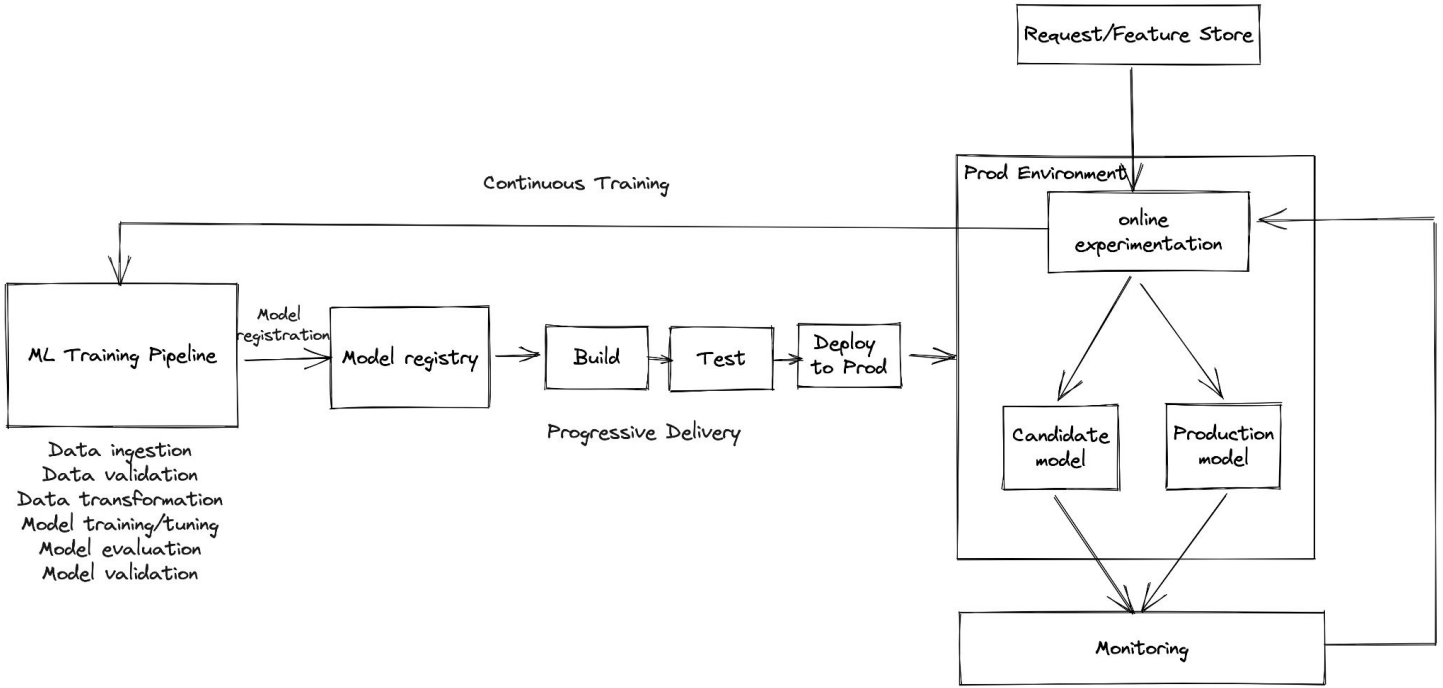
What Is SAP AI Core

Typical user flow in SAP AI Core

1. Users have different AI Scenarios
2. For each AI Scenario, there are workflow templates and serving templates
3. Users train model via workflow template, and save the model into **object store**.
4. Users deploy their model via serving template and object store.

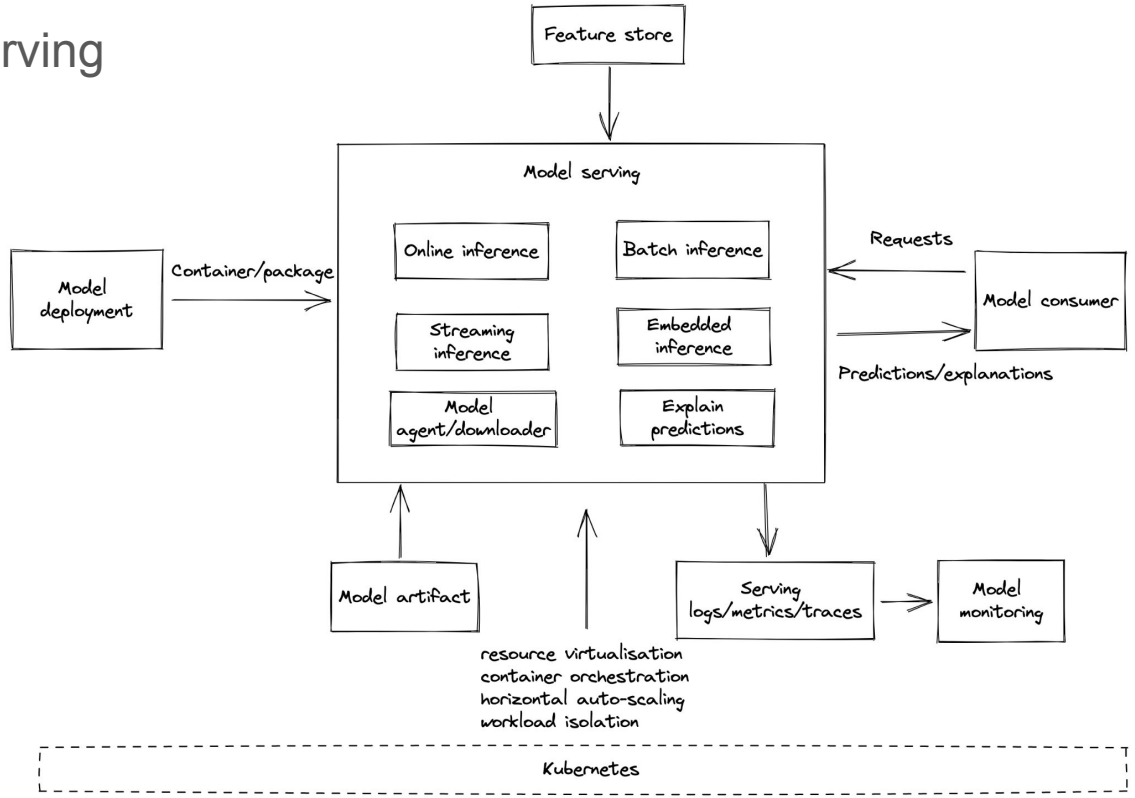
Challenges In ML Model Serving

A Typical ML Pipeline



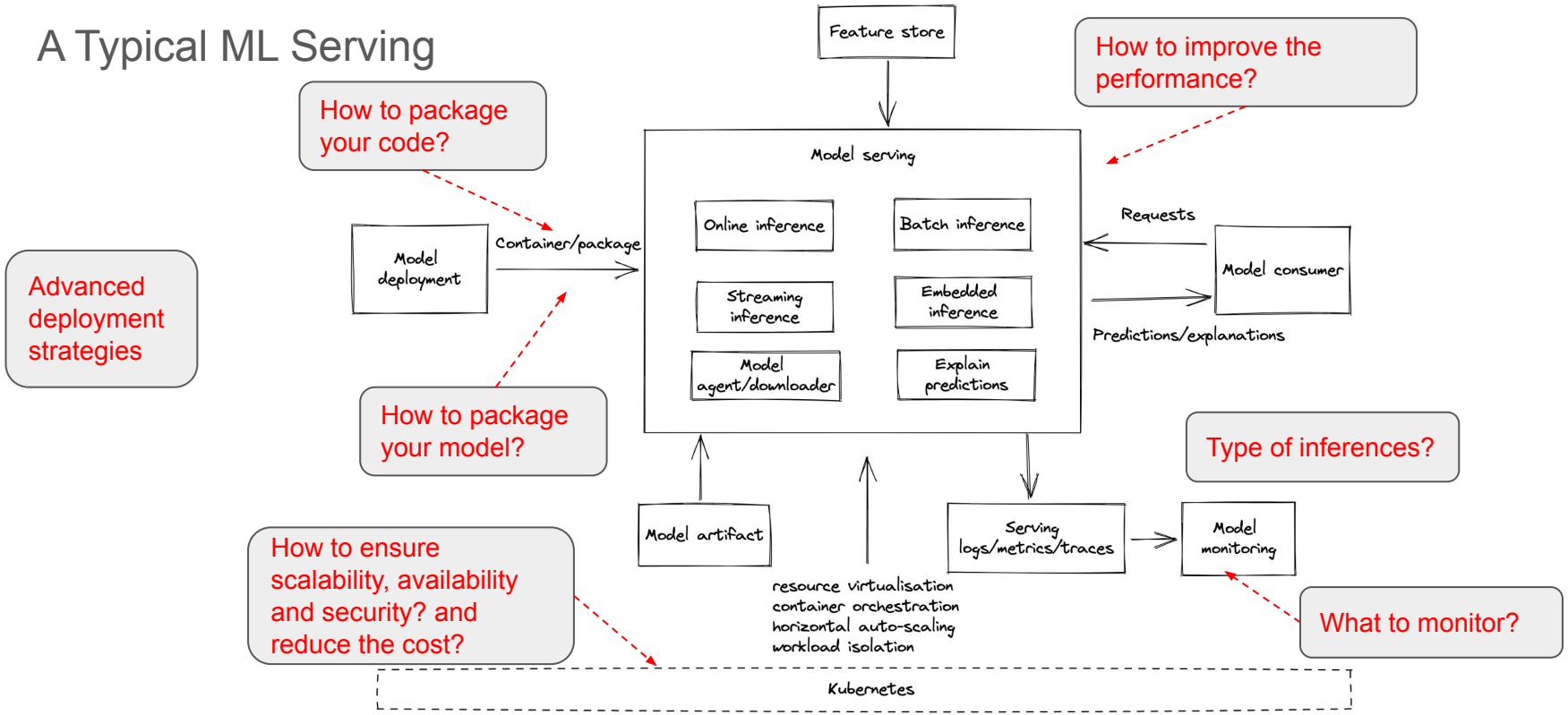
Challenges In ML Model Serving

A Typical ML Serving



Challenges In ML Model Serving

A Typical ML Serving



Challenges in ML Model Serving

Challenges	Solutions
Idle deployment increases the cost	**Serverless** to allow scale to zero for idle deployment
Deployment takes time to scale up	Node pool + image cache, **model cache**, etc

ML workload deployment vs normal workload deployment:

- Compute intensive (CPU/GPU/FPGA etc)
- Image is large (up to 5 GB)
- Model artifact and it is large as well (100 MB ~ 1.5 GB+)

Fluid on Model Artifacts Caching

Background:

- 0 -> 1 scaling is slow
- 1 -> N scaling is slow as well

Why:

- Downloading model artifact takes time (1-2 mins)
- Keep downloading model artifact during the scaling

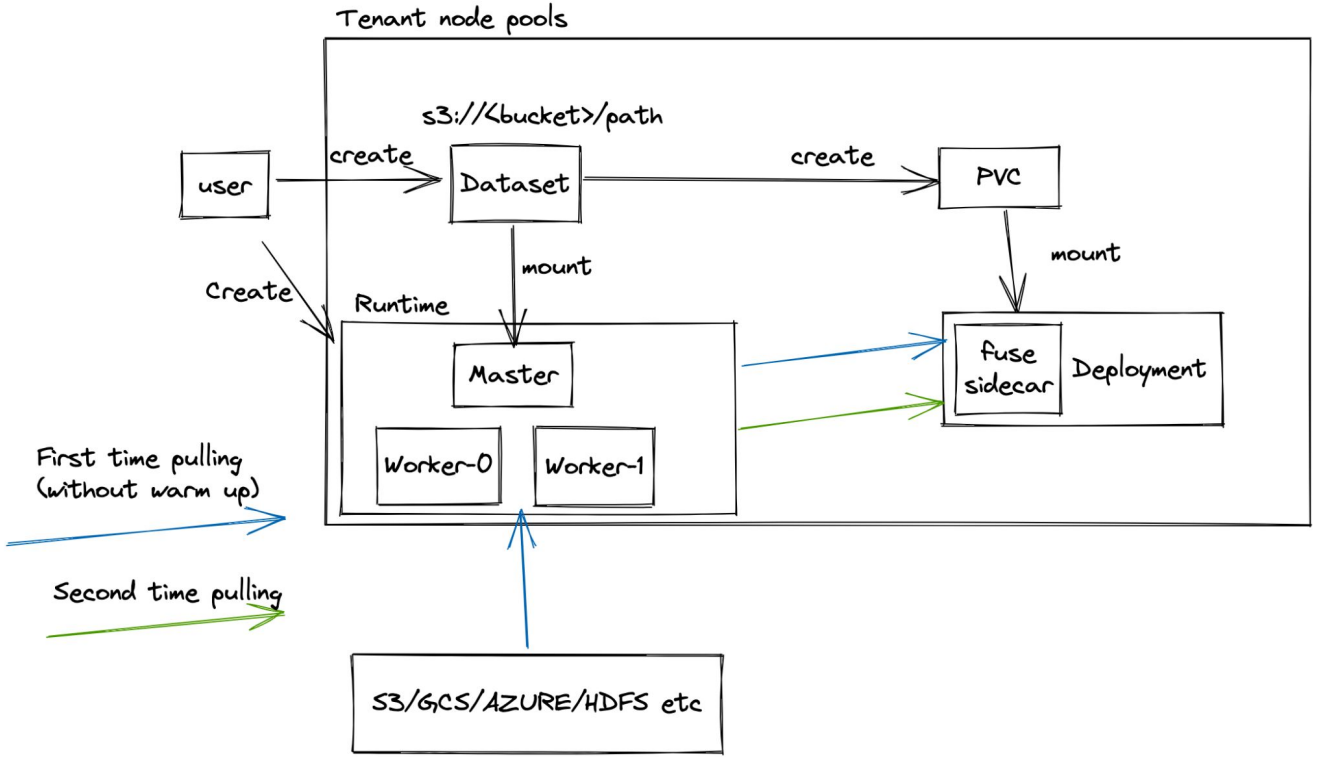
Requirements:

- Fast scaling for ML Workload
- High security on user’s data
- Not add too much cost to the users

Solution:

- Speed up the downloading or preload the model artifact
- Keep the model artifact in the cache, pull the model from the cache rather than remote storage.

Fluid on Model Artifacts Caching



Result and Feedback

- Two worker nodes: m5.large ($0.115/hour or $82/month): vCPU: 2, Memory: 8GiB, Instance Storage: EBS-Only, Network Bandwidth: up to 10 Gbps, EBS Bandwidth: up to 4,750 Mbps. Volume Type gp2, 50GB SSD.
- Model artifact: 1.3 GB model from S3.

1. First time deployment: pull model without cache
2. Second time deployment: pull model from cache

Runtime	Opensource	Language	1st deployment	2nd deployment
Model downloader	YES	Python	1-2 mins	1-2 mins
JindoFS	NO*	C/C++	29 seconds	5 seconds

Result and Feedback

Security

1. Data is within the fuse-sidecar
2. Not able access the data from worker node
3. Once runtime is deleted, data will be removed from the node

Result and Feedback

- Over head in fuse sidecar: creation and deletion is a bit slow
- Fuse sidecar: image is large
- Fuse sidecar: resource usage, can it be a init container?