SELF-STUDY ADDENDUM

SL 4M

Self-Study Addendum — SL 4M Machine Learning Engineer

Palantir Developers Reference Library

HEADQUARTERS
UNITED STATES ARMY EUROPE AND AFRICA
(USAREUR-AF)
Wiesbaden, Germany

26 MARCH 2026

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

SELF-STUDY ADDENDUM — SL 4M MACHINE LEARNING ENGINEER

PALANTIR DEVELOPERS REFERENCE LIBRARY

NOT REQUIRED FOR QUALIFICATION. This addendum provides curated references from the Palantir Developers YouTube channel (@PalantirDevelopers) for personnel who want to deepen their MSS technical skills beyond the core curriculum. All content is publicly available.

Companion Resource — Ontologize Channel: @Ontologize — Official Palantir training partner. 68 indexed video walkthroughs covering Foundry and AIP features. Full catalog with TM cross-references: source_material/ontologize_youtube/README.md

HOW TO USE THIS ADDENDUM

Videos are grouped by topic and ordered from foundational to advanced. Start with the group most relevant to your current work. Videos marked "Core" are the most directly applicable to day-to-day MLE tasks on MSS; videos marked "Advanced" extend into production optimization and platform topics.

GROUP 1 — PYSPARK FUNDAMENTALS

Core skills for writing performant Foundry Transforms. If your feature pipelines are slow or timing out, start here.

Video	What it Covers	Relevant TM Chapter
Spark Basics | Partitions	Spark partition fundamentals — partition count, data skew, and how partitioning affects Transform performance and resource use	Ch 3 (Feature Engineering)

Video	What it Covers	Relevant TM Chapter
Spark Basics	*Shuffling*	Shuffle operations and the cost of wide transformations (groupBy, join, window functions); how to avoid expensive cross-partition data movement	Ch 3 (Feature Engineering)

GROUP 2 — FOUNDRY CODE REPOSITORIES AND TRANSFORM AUTHORING

The core toolchain for writing, testing, and publishing Foundry Transforms. Covers the full development workflow from authoring to production.

Video	What it Covers	Relevant TM Chapter
Code Repositories | *How to Write Data Transformations in Palantir Foundry*	Core procedure for authoring Python Transforms using the `@transform_df` decorator — input/output registration, running in development, and promoting to production	Ch 3 (Feature Engineering)
Code Repositories | *How to Write Incremental Data Transforms in Palantir Foundry*	The `@incremental` decorator pattern for processing only new or changed records; essential for high-frequency MSS data feeds and large feature datasets	Ch 3 (Feature Engineering)
Code Repositories | *How to Use Data Expectations in Palantir Foundry*	Built-in data quality checks using the `@check` decorator; how to validate inputs and outputs programmatically as part of the Transform execution	Ch 3, Ch 7 (MLOps)
Code Repositories | *How to Unit Test PySpark in Palantir Foundry*	Writing and running unit tests for PySpark Transforms inside Foundry Code Repositories; the primary quality gate for ML code before merging to main	Ch 2 (Code Workspaces)
Code Repositories | *How to Create a Python Library in Palantir Foundry*	Creating shared Python libraries in Foundry for reusable logic — relevant when feature engineering patterns are shared across multiple model pipelines	Ch 3 (Feature Engineering)
Code Repositories | *How to Consume a Library in Palantir Foundry*	Importing and using Foundry-hosted Python libraries in Transform code; the consumption side of the library pattern	Ch 3 (Feature Engineering)
Code Repositories | *Development Process and*	End-to-end development workflow: branching, testing, code review, and promotion for production ML pipelines	Ch 2 (Code Workspaces), Ch 3

Video	What it Covers	Relevant TM Chapter
Pipeline Craftsmanship in Palantir Foundry		

GROUP 3 — PIPELINE ARCHITECTURE AND OPTIMIZATION

Advanced patterns for building robust, efficient, and production-grade ML data pipelines.

Video	What it Covers	Relevant TM Chapter
Deep Dive: Optimizing Data Pipelines with Iceberg Tables and Lightweight Compute | DevCon 4	Iceberg table format for time-travel, snapshot isolation, and efficient incremental reads; lightweight compute patterns that reduce pipeline overhead for production ML workloads	Ch 7 (MLOps)
Chad & Xander | Lightweight Transforms in Pipeline Builder	Lightweight transform patterns in Foundry Pipeline Builder that reduce compute cost for simple transformations that do not require full Spark	Ch 3 (Feature Engineering)
Chad & Nicolas | Lightweight Transforms at Merck KGaA, Darmstadt, Germany	Case study: enterprise-scale adoption of lightweight transforms for data pipeline efficiency; pattern library from a large production Foundry deployment	Ch 3 (Feature Engineering)
Chad & Matt | Lightweight Data Transforms with Palantir AIP	Lightweight transform patterns in the context of AIP-integrated pipelines; how to keep data movement efficient when ML outputs feed AIP workflows	Ch 6 (Model Deployment)
Schedules | Creation, Configuration, and Execution in Palantir Foundry	How to create, configure, and run Foundry Schedules for feature pipelines and inference Transforms; covers schedule dependencies and execution history	Ch 7 (MLOps)
Schedules | Management, Metrics, and Triggers in Foundry	Managing schedules at scale: monitoring execution health, interpreting schedule metrics, and configuring event-based triggers for pipeline automation	Ch 7 (MLOps)
Foundry Reference Project | Data Pipeline	Foundry's reference pipeline implementation: an end-to-end example covering ingestion, transformation, and output patterns in a production-representative project	Ch 3, Ch 7
Foundry Reference Project | Structure	Reference project structure and repository layout conventions for Foundry ML and data engineering projects	Ch 2 (Code Workspaces)

GROUP 4 — MLOPS AND MONITORING

Pipeline health monitoring, drift detection, and production observability — the operational disciplines that keep production models reliable.

Video	What it Covers	Relevant TM Chapter
Pipeline Monitoring | How to Start Monitoring Data Health in Palantir Foundry	Foundational pipeline health monitoring in Foundry: setting up dataset checks, health indicators, and monitoring dashboards for a single dataset or transform	Ch 7 (MLOps)
Pipeline Monitoring | How to Monitor Health Across a Pipeline in Palantir Foundry	Extending monitoring to full pipeline visibility: tracking data quality and freshness across a chain of dependent Transforms; identifying and alerting on pipeline-level failures	Ch 7 (MLOps)

GROUP 5 — COMPUTE MODULES AND REAL-TIME INTEGRATION

Covers on-demand and streaming inference patterns, and the AIP Compute Module integration for embedding ML capability in AIP workflows.

Video	What it Covers	Relevant TM Chapter
Build with AIP: Compute Modules	Compute Modules — how to run custom code (including ML model inference) within AIP Logic workflows; the integration point between MLE-built models and AI Engineer-built AIP products	Ch 6 (Model Deployment)
AIP with Jeg: Building a Streaming Ingestor with Compute Modules	Building a real-time streaming data ingestor using Compute Modules; relevant when the operational requirement requires sub-minute data freshness feeding ML inference	Ch 6 (Model Deployment)
Applied AI: Scaling AI Workflows and Task Execution with AIP	How AIP workflows scale across complex, multi-step task execution pipelines; the AI Engineer perspective on consuming ML model outputs — useful for understanding the design requirements for model endpoints and output schemas	Ch 6 (Model Deployment)

GROUP 6 — PLATFORM AND ONTOLOGY REFERENCE

Background on Foundry platform architecture and the Ontology layer that ML models publish into.

Video	What it Covers	Relevant TM Chapter
Palantir Ontology Overview	Conceptual overview of the Foundry Ontology — Object Types, properties, and relationships. Provides the platform context for deploying model outputs as Ontology Object properties	Ch 6 (Model Deployment)
Product Launch: Media, Real-Time Updates, and Expressive Compute in OSDK | DevCon 2	Platform SDK capabilities including real-time updates and expressive compute patterns; background on how OSDK applications consume model outputs from the Ontology	Ch 6 (Model Deployment)

GROUP 7 — CASE STUDIES AND ADVANCED TOPICS

Real-world implementations and DevCon presentations that illustrate advanced ML platform and MLOps concepts.

Video	What it Covers	Relevant TM Chapter
xAI x TWG for Model Tuning Infrastructure | DevCon 2	Infrastructure decisions for model fine-tuning at scale: compute allocation, checkpoint management, and training-serving integration in a production platform	Ch 4 (Model Training)

USAREUR-AF Operational Data Team