

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

SELF-STUDY ADDENDUM

SL 5M



Self-Study Addendum — SL 5M Advanced Machine Learning Engineer

Palantir Developers Reference Library

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

DRAFT — NOT FOR OFFICIAL USE. FOR TRAINING PLANNING PURPOSES ONLY.

26 MARCH 2026

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

SELF-STUDY ADDENDUM — SL 5M ADVANCED MACHINE LEARNING ENGINEER

PALANTIR DEVELOPERS REFERENCE LIBRARY

NOT REQUIRED FOR QUALIFICATION. This addendum provides curated references from the Palantir Developers YouTube channel ([@PalantirDevelopers](#)) for senior MLEs who want to extend their platform knowledge beyond the core curriculum. All content is publicly available. SL 5M personnel are expected to have already worked through the SL 4M addendum; this addendum extends it with advanced platform, architecture, and enterprise-scale topics.

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 organized from foundational (inherited from SL 4M scope) through advanced (SL 5M platform and architecture focus). Senior MLEs should prioritize Groups 3–8, which address the platform design, enterprise scale, and cross-platform integration topics central to SL 5M responsibilities. Groups 1–2 are retained from the SL 4M addendum for completeness.

GROUP 1 — PYSPARK FUNDAMENTALS

(Inherited from SL 4M addendum — included for completeness. SL 5M MLEs should be fluent in this material.)

Video	What it Covers	Relevant TM Chapter
<i>Spark Basics Partitions</i>	Spark partition fundamentals — partition count, data skew, and impact on Transform performance	Ch 2–3 (background)

Video	What it Covers	Relevant TM Chapter
<i>Spark Basics Shuffling</i>	Shuffle operations and how to avoid expensive wide transformations	Ch 2–3 (background)

GROUP 2 — FOUNDRY CODE REPOSITORIES AND TRANSFORM AUTHORIZING

(Inherited from SL 4M addendum. SL 5M MLEs set code review standards for this toolchain.)

Video	What it Covers	Relevant TM Chapter
<i>Code Repositories How to Write Data Transformations in Palantir Foundry</i>	Core Transform authoring procedure — the baseline all MLEs reviewed in SL 4M	Ch 8 (code review standards)
<i>Code Repositories How to Write Incremental Data Transforms in Palantir Foundry</i>	<code>@incremental</code> pattern for efficient incremental processing	Ch 2 (automated retraining pipelines)
<i>Code Repositories How to Use Data Expectations in Palantir Foundry</i>	<code>@check</code> decorator for programmatic data quality gates in Transforms	Ch 2 (data validation stage)
<i>Code Repositories How to Unit Test PySpark in Palantir Foundry</i>	Unit testing PySpark Transforms inside Foundry — the minimum standard for all production ML code	Ch 8 (code review standards)
<i>Code Repositories How to Create a Python Library in Palantir Foundry</i>	Creating shared Python libraries in Foundry; pattern for canonical feature engineering logic in a shared feature store	Ch 8 (shared feature store)
<i>Code Repositories How to Consume a Library in Palantir Foundry</i>	Consuming Foundry-hosted Python libraries in Transform code	Ch 8 (shared feature store)
<i>Code Repositories Development Process and Pipeline Craftsmanship in Palantir Foundry</i>	End-to-end development workflow and pipeline craftsmanship standards	Ch 8 (code review, team development)

GROUP 3 — PIPELINE ARCHITECTURE AND OPTIMIZATION

Production-grade pipeline design for the ML platform infrastructure SL 5M MLEs design and maintain.

Video	What it Covers	Relevant TM Chapter
<i>Deep Dive: Optimizing Data Pipelines with Iceberg Tables and Lightweight Compute DevCon 4</i>	The primary DevCon deep dive on Iceberg table optimization: time-travel, snapshot isolation, efficient incremental reads, and lightweight compute. Directly relevant to feature store versioning, training dataset snapshotting, and production pipeline architecture	Ch 2 (automated retraining), Ch 8 (feature store)
<i>Chad & Xander Lightweight Transforms in Pipeline Builder</i>	Lightweight transform patterns that reduce Spark overhead for simple feature computations; a design option for canonical feature store Transforms where full Spark is not warranted	Ch 8 (shared feature store)
<i>Chad & Nicolas Lightweight Transforms at Merck KGaA, Darmstadt, Germany</i>	Enterprise case study: large-scale production adoption of lightweight transforms and the governance model around standardizing pipeline patterns	Ch 8 (platform architecture)
<i>Chad & Matt Lightweight Data Transforms with Palantir AIP</i>	Lightweight transform patterns in AIP-integrated pipelines; keeping data movement efficient when ML outputs feed downstream AIP workflows	Ch 8 (cross-track integration)
<i>Schedules Creation, Configuration, and Execution in Palantir Foundry</i>	Creating, configuring, and executing Foundry Schedules for automated pipeline orchestration	Ch 2 (automated retraining pipelines)
<i>Schedules Management, Metrics, and Triggers in Foundry</i>	Managing schedules at scale; execution metrics and event-based trigger patterns for automated retraining pipelines	Ch 2 (automated retraining pipelines)
<i>Foundry Reference Project Data Pipeline</i>	Foundry reference end-to-end pipeline: production-representative patterns for ingestion, transformation, and output	Ch 2, Ch 8
<i>Foundry Reference Project Structure</i>	Reference project structure and repository layout conventions; the standard SL 5M MLEs should enforce for ML program repositories	Ch 8 (code review, team development)

GROUP 4 — MLOPS AND MONITORING

Platform-level monitoring infrastructure design — beyond individual pipeline health to enterprise ML observability.

Video	What it Covers	Relevant TM Chapter
<i>Pipeline Monitoring How to Start Monitoring Data Health in Palantir Foundry</i>	Foundational monitoring setup: dataset checks, health indicators, and monitoring dashboards	Ch 2 (production readiness gates)
<i>Pipeline Monitoring How to Monitor Health Across a Pipeline in Palantir Foundry</i>	Full pipeline health visibility across chains of dependent Transforms; the pattern for monitoring automated retraining pipeline stages	Ch 2 (automated retraining pipelines)

GROUP 5 — COMPUTE MODULES AND REAL-TIME INTEGRATION

On-demand and streaming inference patterns; the AIP integration interface that SL 5M MLEs design the platform side of.

Video	What it Covers	Relevant TM Chapter
<i>Build with AIP: Compute Modules</i>	Compute Modules for running custom code inside AIP Logic workflows; the interface design contract between the ML platform and AIP products	Ch 8 (cross-track integration: SL 4H)
<i>AIP with Jeg: Building a Streaming Ingestor with Compute Modules</i>	Streaming data ingestor pattern using Compute Modules; relevant for real-time inference architecture design in Chapter 6	Ch 6 (real-time inference)
<i>Applied AI: Scaling AI Workflows and Task Execution with AIP</i>	How AIP workflows scale across complex multi-step task execution; the AI Engineer's operational perspective on consuming ML model outputs — essential context for designing the model endpoint interface contract	Ch 8 (cross-track integration: SL 4H)

GROUP 6 — PLATFORM AND ONTOLOGY REFERENCE

Foundry platform architecture and the OSDK layer that production ML systems publish into.

Video	What it Covers	Relevant TM Chapter
<i>Palantir Ontology Overview</i>	Conceptual Ontology overview — Object Types, properties, relationships. Platform context for the model output publication patterns in SL 4M and SL 5M	Ch 8 (platform architecture)
<i>Product Launch: Media, Real-Time Updates, and Expressive Compute in OSDK DevCon 2</i>	OSDK capabilities: real-time updates and expressive compute patterns; background on how downstream applications consume model outputs	Ch 6 (real-time inference), Ch 8 (cross-track integration)

GROUP 7 — CASE STUDIES AND ENTERPRISE ARCHITECTURE

Real-world DevCon presentations illustrating advanced ML platform design, enterprise deployment, and organizational ML capability.

Video	What it Covers	Relevant TM Chapter
<i>xAI x TWG for Model Tuning Infrastructure DevCon 2</i>	Infrastructure architecture for model fine-tuning at scale: compute allocation, checkpoint management, and training-serving integration in a production platform context	Ch 3 (advanced neural architectures)

GROUP 8 — ADVANCED PLATFORM AND ENTERPRISE SCALE (SL 5M SPECIFIC)

DevCon 5 content focused on enterprise-scale ML platforms, interoperability, and organizational AI capability. Targeted at SL 5M MLEs designing the platform, not operating within it.

Video	What it Covers	Relevant TM Chapter
<i>Deep Dive: Interoperability at Scale with the Multimodal Data Plane DevCon 5</i>	How Palantir's data plane handles interoperability across heterogeneous data types and systems at enterprise scale; relevant to coalition data sharing architectures and cross-domain ML scenarios in Chapter 4	Ch 4 (federated and privacy-preserving ML)
<i>Product Launch: Developer Experience DevCon 5</i>	New developer experience capabilities in Foundry; platform improvements relevant to ML workflow design, code repository tooling, and the MLE development experience at scale	Ch 8 (platform architecture, team development)
<i>Akshay Krishnaswamy Opening Remarks DevCon 5</i>	Platform direction and strategic priorities from Palantir leadership; context for understanding where Foundry's ML capabilities are heading and how to align MSS platform architecture decisions with product trajectory	Ch 1 (scope and platform context)
<i>Building Enterprise Autonomy with Shyam Sankar, CTO</i>	Strategic framing of enterprise AI and autonomous workflows from Palantir's CTO; relevant to SL 5M MLEs who advise leadership on ML platform direction and the role of automated systems in operational decision chains	Ch 1, Ch 2 (automated MLOps governance)

USAREUR-AF Operational Data Team