# Od inferenčního serveru k produkci
# Jak bezpečně škálovat AI workload

Lukáš Smiga · Cloud & Middleware Evangelist, ELOS Technologies
Red Hat Innovate Together, Praha, 6. 10. 2026

# Výchozí bod: dedikovaný inferenční server

## Co funguje

- Modely běží na dedikovaném serveru s Red Hat Enterprise Linux AI.

## Co chybí

- Vysoká dostupnost
- Jednotné přihlášení a role
- Centrální logování
- Oddělení testu a produkce

## Jakou AI platformu znali

- Postavenou na konzervativních principech.
- Příznaky v datových sadách se označují ručně.

# Co chce tým datových vědců v regulované instituci

## Data

- **Kde leží,**
- **jak se verzují,**
- **jak vypadají štítky a jejich definice.**

## Nástroje

- **Experimenty z VS Code nebo PyCharmu,**
- **sledování a porovnání modelů.**

## Provoz

- **Test a produkce zvlášť,**
- **denní dávková úloha i trvale běžící API.**

## Audit

- **Zdraví modelu po nasazení,**
- **přetrénování,**
- **důkazy pro AI Act.**

*Nechtějí jen přesný model. Chtějí vědět, z jakých dat vznikl, kdo ho schválil a jak ho zopakovat.*

# Stejný motor, nová platforma

## Na serveru

- **Red Hat Enterprise Linux AI: inferenční platforma pro jeden server.**
- **Red Hat AI Inference (vLLM).**

## Co se přenáší

- **Stejný inferenční engine vLLM.**
- **Stejné validované modely, balené jako OCI image.**

## Na platformě

- **Red Hat OpenShift AI: KServe s vLLM runtime.**
- **Model z OCI image, ze S3 nebo z PVC.**




Zdroj: Red Hat

***Nepřepisujeme, přestěhujeme. Platforma přidá to, co jeden server nedá.***

# Co OpenShift přidává k inferenčnímu serveru

| | |
|---|---|
| Bezpečný základ | **Neměnné uzly RHEL CoreOS, Security Context Constraints, NetworkPolicy** |
| Řízení | **Registr a katalog modelů s verzemi a povýšením, RBAC, kvóty týmů** |
| Vývoj a test | **Workbenche, MLflow, pipelines, oddělené projekty pro test a produkci** |
| Provoz | **KServe a vLLM, Models-as-a-Service, autoscaling, Kueue** |
| Pozorovatelnost | **Dashboardy i pro neadministrátory, metriky TTFT a TPOT, hodnocení přes EvalHub** |

# Datoví vědci si nesou své nástroje

## Vlastní počítač

**Lokální VS Code, data a výpočet na vlastním stroji.**

## Lokálně, napojeno na cluster

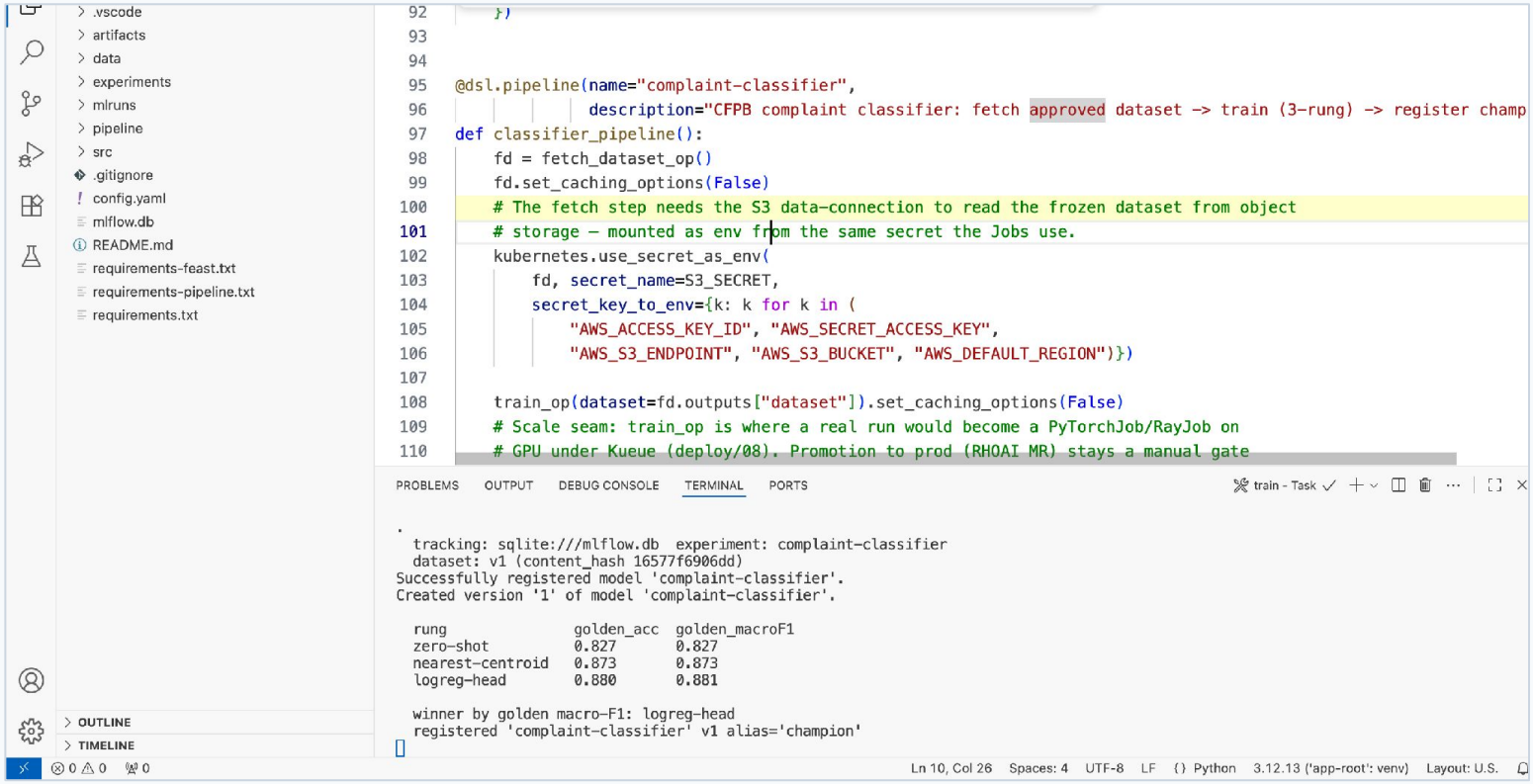
**Lokální kód, ale MLflow, registr a úložiště na platformě.**

## VS Code v podu

**Desktopové VS Code připojené k workbenchi na clusteru. Bez Jupyteru.**

## Pipeline

**Stejný kód jako krok Data Science Pipeline v clusteru.**




VS Code jako workbench v Red Hat OpenShift AI (Developer Sandbox)

*Čtyři způsoby práce, stejný výsledek: shodné metriky i shodný otisk dat.*

# Příznaky dnes počítá pomocný model

## Bez označených dat

- **Zero-shot: kategorie popsaná větou, model hledá podobnost.**
- **Makro-F1 0,827**

## Bez tréninku

- **Nejbližší centroid z převzatých štítků.**
- **Makro-F1 0,873**

## Lehký trénink

- **Logistická regrese nad zmrazenými embeddingy.**
- **Makro-F1 0,881**

***Příznaky jsou embeddingy z předtrénovaného modelu, nikdo je neznačí ručně. Veřejná data (stížnosti CFPB), 3 kategorie, kontrolní sada 150 příkladů. Jen CPU.***

# Řízená cesta z vývoje do produkce

Schválený základní model se stahuje z interního úložiště, ne z internetu.

| | |
|---|---|
| Data | **Zmrazená datová sada s otiskem. Feast drží příznaky a štítky.** |
| Vývoj | **MLflow: vývoj a porovnání modelů** |
| Schválení | **Ruční schválení, pak Registr modelů: produkce** |
| Provoz | **Denní CronJob: klasifikuje dávku a skončí** |
| Lidský dohled | **Nejistý výsledek jde člověku** |

# Nudný model vyhrává

01 **Pětkrát větší embeddingový model přidal natrénované hlavě jen 2,6 bodu F1.**

02 **Vyzývatel (MLP) remizoval se šampionem. Šampion zůstává.**

03 **Povýšení modelu je vědomé rozhodnutí člověka, ne automat.**

04 **Nový model převezme další běh úlohy bez přestavby image.**

05 **Závislost na internetu za běhu: žádná.**

# Škálování: od jednoho serveru ke sdíleným GPU

## Serving

**KServe s vLLM runtime pro NVIDIA, AMD, Intel Gaudi i CPU.**

## Chytré směrování

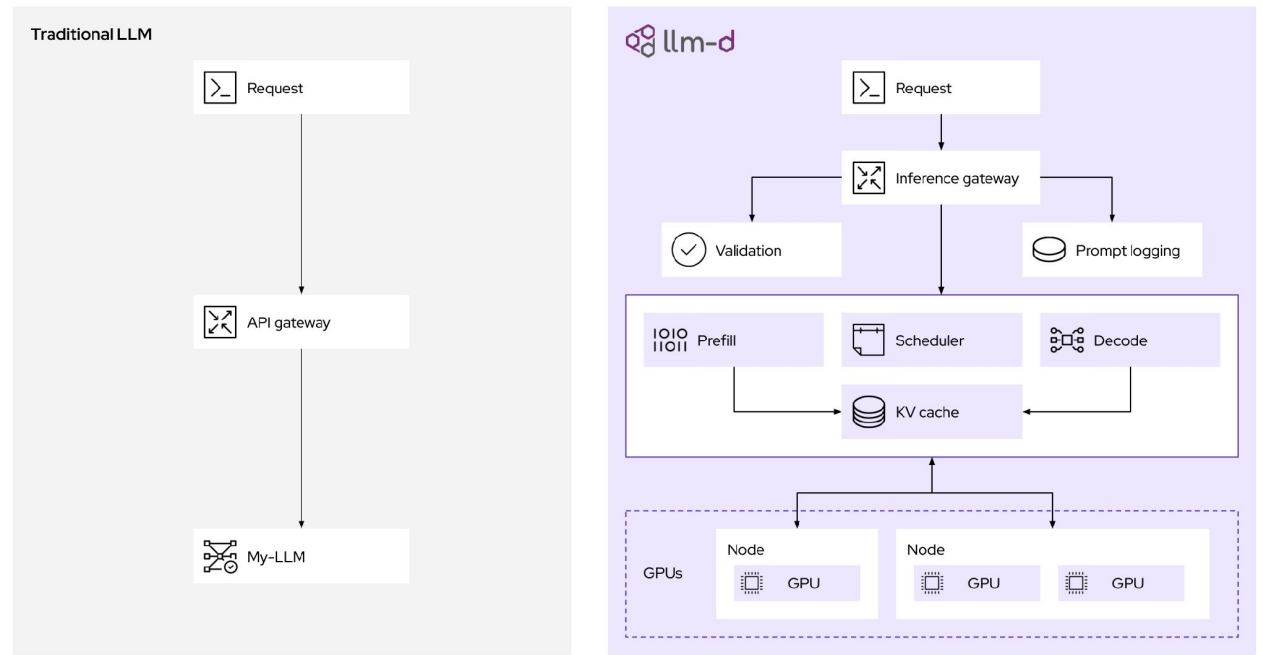
**llm-d posílá požadavek na repliku, která už má prefix v cache.**

## Férové GPU

**Kueue: kvóty týmů, půjčování kapacity, priority.**

## AI jako služba

**Models-as-a-Service: API klíče, kvóty tokenů, limity.**




Zdroj: Red Hat

*OpenShift AI 3.5 (září 2026): obsluha s ohledem na priority a férové plánování GPU.*

# Bezpečně: žádná vrstva sama nestačí

01 **Izolace: neměnné uzly, SCC, NetworkPolicy, oddělené projekty.**

02 **Dodavatelský řetězec: SBOM, SLSA, podpisy Sigstore.**

03 **Podepsané ModelCary od Red Hatu, základní kontrola zranitelností.**

04 **Behaviorální testy modelu: EvalHub, garak, automatizovaný red teaming.**

05 **Guardrails za běhu a člověk u nejistých výsledků.**

# Co platforma nedá sama a musíte si to dostavět

## Data

- **Verzování dat ve stylu gitu.**
- **Otisk a zmrazení dnes řešíme sami.**

## Dohled

- **Sledování driftu a zdraví modelu je potřeba nastavit a napojit.**

## Regulace

- **Dokumentaci pro AI Act platforma nenapíše.**
- **Dá vám k ní stopu.**

***A otázka, kterou má každý: jak AI nástroje bezpečně pustit k interním datům.***

# Otázky?

# Díky!

elos.navy

Lukáš Smiga, ELOS Technologies

elos.navy