Why Use Kubernetes For AI?

Red Hat User Group

Brian Ball
Associate Principal AI
Specialist Solutions Architect

Red Hat

Agenda

1. Why Kubernetes?
2. Personal Anecdotes
3. MLOps as DevOps

Red Hat

Why Kubernetes?

Red Hat

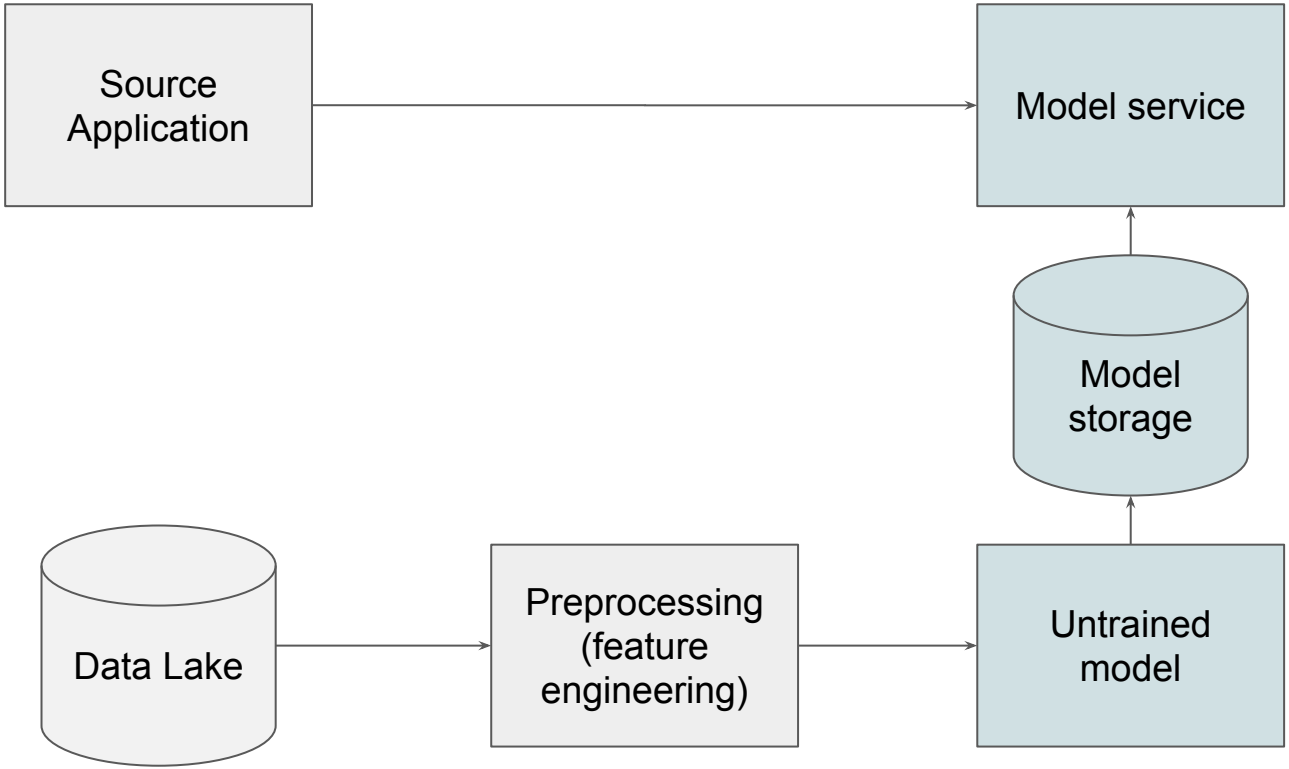
AI Is Complicated

- AI requires many moving parts spanning multiple disciplines
- Very few platforms can handle all of the necessary components to run a production system

- Kubernetes is one such platform!

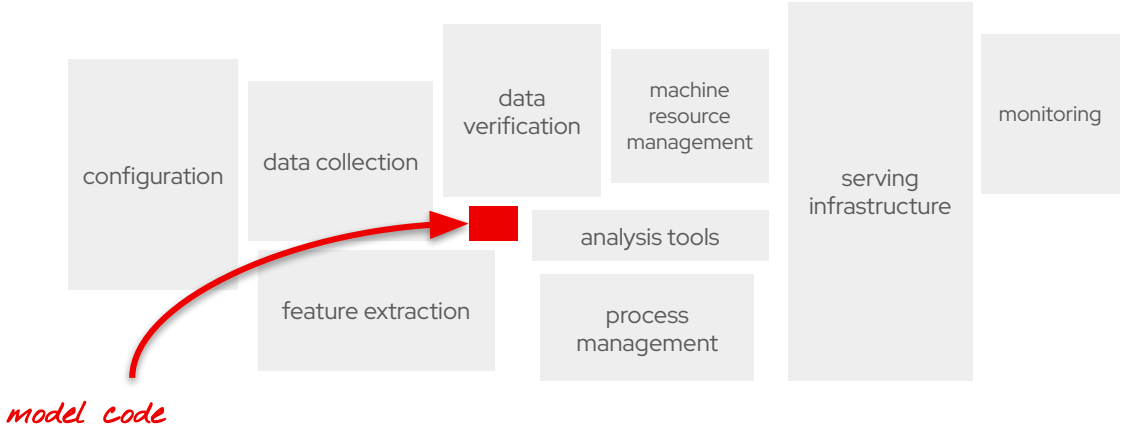
Red Hat

Basic Model Process



Red Hat

Model code is one component of a larger system



Hidden Technical Debt in Machine Learning Systems

"Only a small fraction of real-world ML systems is composed of the ML code...The required **surrounding infrastructure is vast and complex.**"

"Developing and deploying ML systems is relatively fast and cheap, but **maintaining them over time is difficult and expensive**"

Adapted from Sculley et al., "Hidden Technical Debt in Machine Learning Systems." NIPS 2015

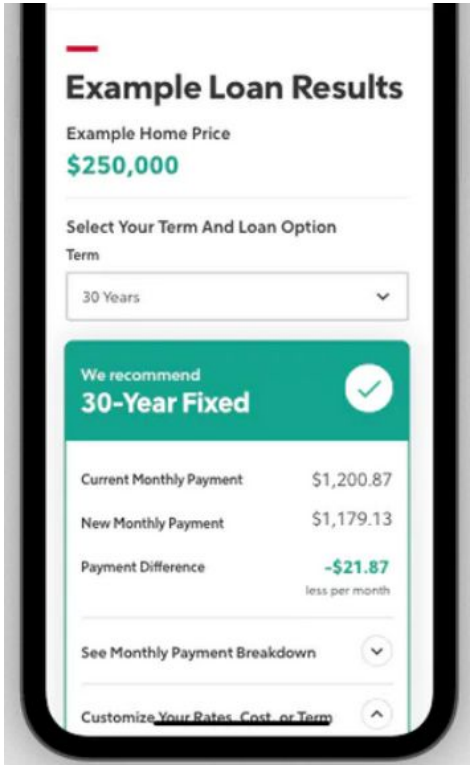
Red Hat

Personal Anecdotes

Red Hat

Mortgage Recommendation Engine

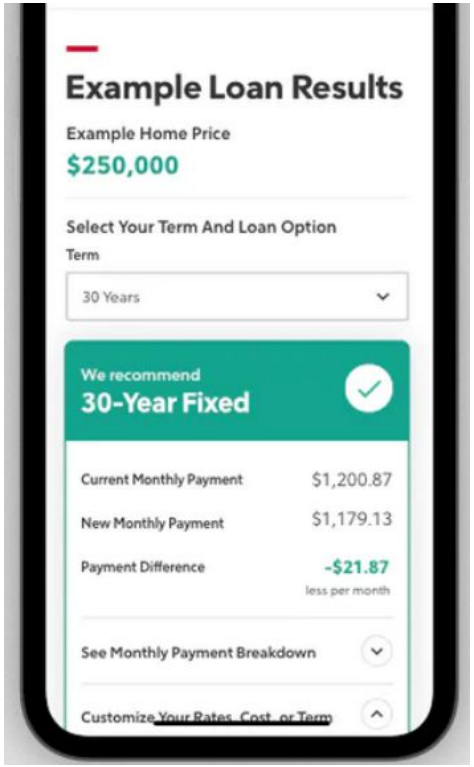
- 100% online mortgage application platform
- Base application uses financial math to determine every eligible mortgage solution for the customer



Red Hat

Mortgage Recommendation Engine

- **Our task was to build a system that chose the best one option to present**
  - **Target to improve over the base rules system**



Red Hat

Mortgage Recommendation Engine

- Not a normal recommender system
  - Options are unique for the financial situation
  - Can only present and choose one option
- 3 components:
  - Baseline
  - Weights
    - Dependent on customer financial characteristics
  - Benefits
    - Dependent on product specifics
    - In an ideal world, these are monotonic

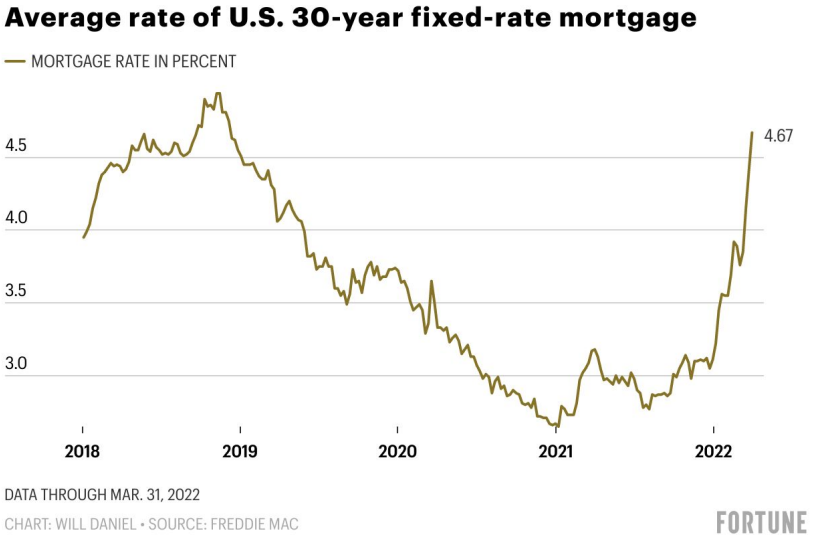
Red Hat

Mortgage Recommendation Engine

- Tight latency SLA
  - Partially evaluate model, then pass back DSL to evaluate in the application
- Because we're making a decision where humans are involved, we have to evaluate models live
  - Don't know how someone is going to react to a recommendation until we make it
  - Predictive accuracy is meaningless because we're trying to drive a reaction which can happen even if the customer doesn't pick our recommendation
  - Models get A/B tested with test fractions changing regularly. New models trained weekly

Red Hat

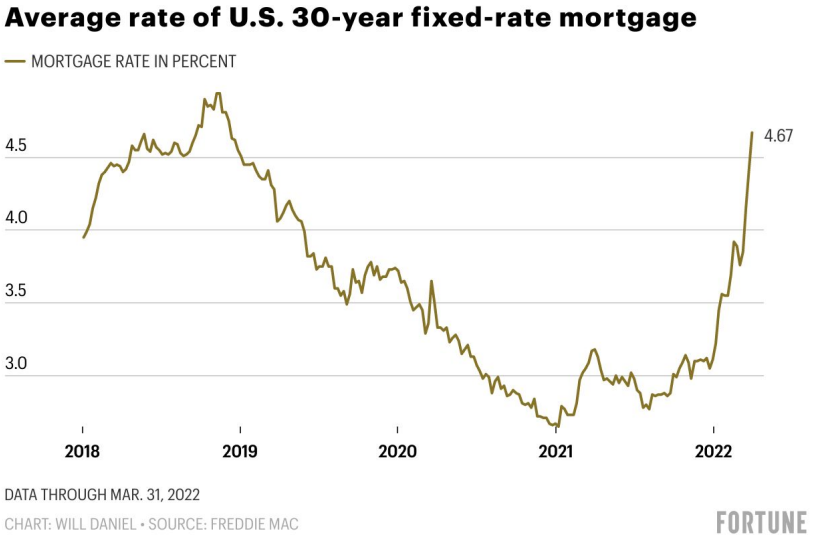
Intraday Lock Volume



- Mortgage broker needs to set prices in the middle of the day to attract more business
- Higher prices get passed to the customer in the form of higher interest rates
  - Lenders are incentivized to go with brokers of the lowest price
- This was late 2022, when the housing market was actively crashing

Red Hat

Intraday Lock Volume



- **Our task was to predict how much loan volume we would generate that day**
  - **To within 20% accuracy**
  - **Add uncertainty bands to all predictions**
  - **Want ability to run simulations based on changing prices at certain times in the future**

Red Hat

Intraday Lock Volume

- Prediction depends on several factors:
  - Product Line
  - Historical trends
    - Seasonal
    - Day of week
    - Time of day
      - Time zones
  - Volume already seen that day (autoregression)
  - Internal and market pricing
- What ended up mattering for pricing was *by how much we changed it*

Red Hat

Intraday Lock Volume

- Accuracy changes throughout the day
  - Can't train against all times of day because the data is correlated
    - Training up to 4 PM uses all of the same data and more than training up to 2 PM
  - Just monitored it to make sure we stayed close
- Something has to pull data and aggregate it specifically for this version of the model
  - Need to keep this service in sync with the model
- Build single service to handle custom model structure and data orchestration
  - Rotate models as they are trained and evaluated

Red Hat

Algae Weight Model

- Carbon capture by sinking bio-material in the deep ocean
- Need to quantify carbon captured, environmental impact, etc.



Red Hat

Algae Weight Model

- Can’t directly measure effects, but we can deploy sensors
  - GPS
  - Cameras
- Build scientific models from the sensor data to quantify



Red Hat

Algae Weight Model

- **Our task was to quantify carbon in a deployment from images of growing macroalgae**



Red Hat

Algae Weight Model

- Train and validate a model in the lab
  - Hope that this matches what happens in the ocean (it doesn't)
- Classify out images of things that weren't supposed to happen
  - E.g. fish, torn mesh cage
- From pictures, need to segment what is macroalgae and what is not
- No intuition on what image effects would predict carbon content so we had to try things.
  - The darker green and blue the macroalgae is linearly correlates with the amount

Red Hat

Algae Weight Model

- End goal to run the classification on the buoy and retake bad images before saving
- Chain models to get the final prediction
  - Segment the algae
  - Build features from segmented image
  - Predict the carbon content
- Built single repository with multiple deployables to handle sequential training of models and chained execution

Red Hat

“Nonstandard” Model Components

Recommender System

- Model must be deployed and used to be evaluated

Intraday Lock Volume

- Model requires real-time data orchestration to work

Algae Weight Model

- Model is chain of subprocesses and models

MLOps as DevOps

Red Hat

AI Isn’t Just A Model

- AI are data intensive applications that require the right data to work
- Strict data contracts
  - Software
  - Mathematical
- Models are static in dynamic systems
  - Grow stale over time as the system changes
  - Can retrain the same code with new data and get a different model

Red Hat

AI Is A Full System

- AI requires multiple components in production
  - Data cleaning
  - Training pipelines
  - Serving environment + orchestration
  - Monitoring and Alerting
- There is a lot to automate
  - CI/CD Pipelines
  - Enterprise Integration
    - Even if you're using an AI platform, you still have to integrate that with other systems to work
  - Model Retraining and Rotation

Red Hat

Software Services

- Modern Software Engineering follows DevOps and Service Oriented Architecture
- You should adopt those practices for your AI systems
- DataOps is moving towards DevOps, and that applies too
- Kubernetes is a powerful tool for DevOps

Red Hat

How To Think About Your AI

- Think about your problem to determine the right approach
- Think about the entire system
  - Remember that training and serving must match
  - Your model can be abstracted like any other programming function
- Treat the data cleaning as part of your model
  - Data cleaning often needs to be part of your operational service
- Treat your model artifact like data

Red Hat

Questions?



linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHat

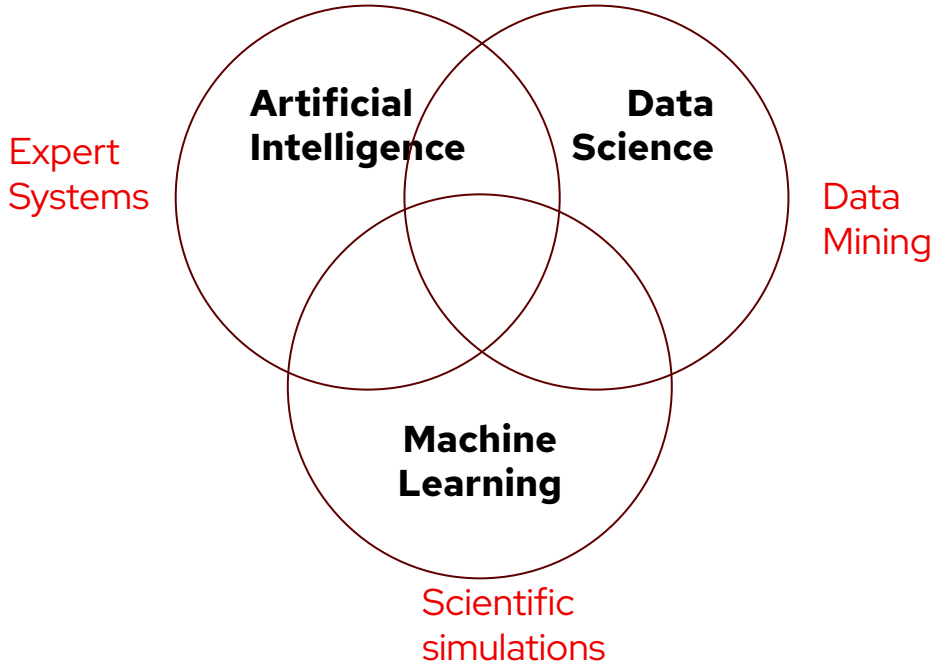
Red Hat

Extra

Red Hat

What is AI?

Artificial intelligence is a field of science concerned with building computers and machines that can reason, learn, and act in such a way that would normally require human intelligence or that involves data whose scale exceeds what humans can analyze.



https://cloud.google.com/learn/what-is-artificial-intelligence?hl=en

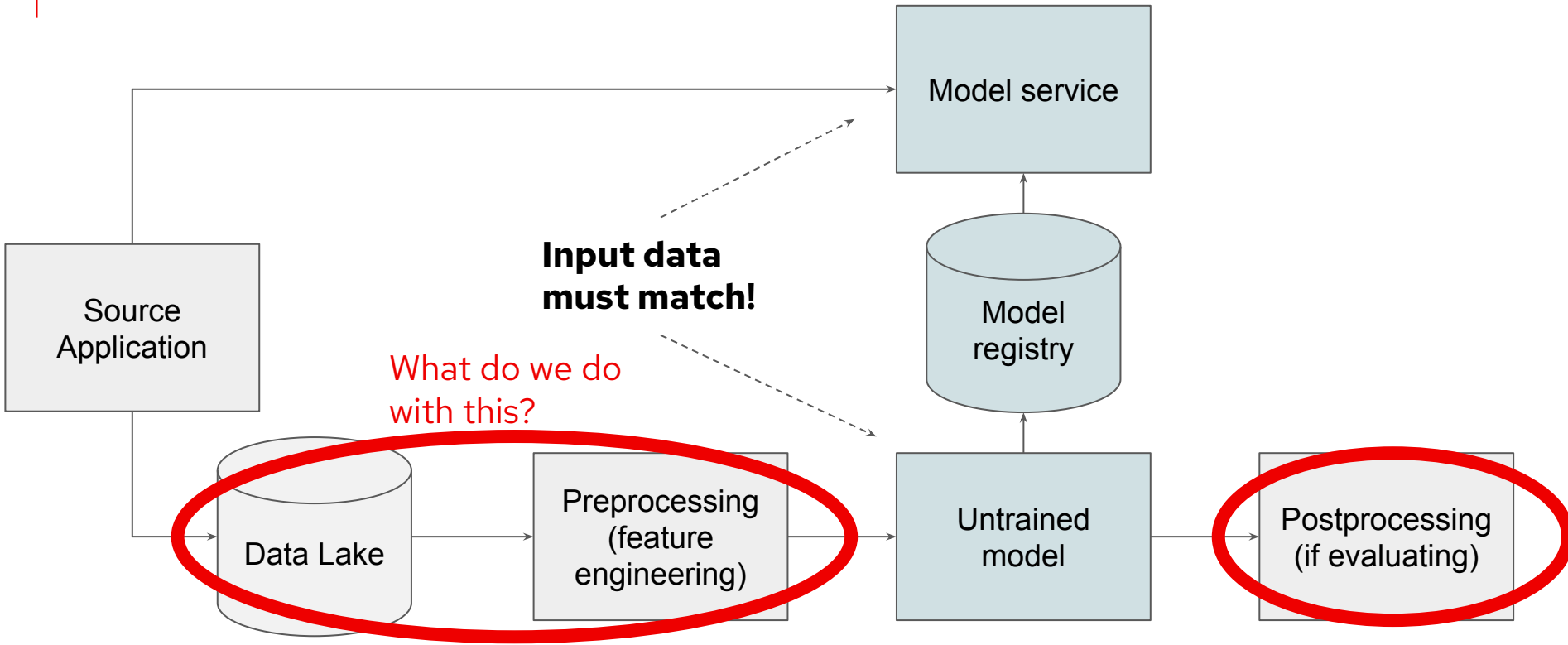
Red Hat

Retrieval Augmented Generation

- Supplement your large language model (LLM) with specific knowledge data
- Store and retrieve data from extra database
  - Use an embedding model to know where to pull from your database
  - Then process using the LLM
- Functionally identical to the data orchestration from the Intraday Lock Volume project

Red Hat

Basic Model Process



Red Hat