PREFECT

CASE STUDY

Inside Baseball: How The Nationals Win With Analytics

By automating a wide range of data processes, the team can devote more time to business-critical objectives: improving the performance of their players and setting a course for team development.

COMPANY

INDUSTRY

Professional Sports

HEADQUARTERS

Washington, DC

PRODUCTS USED

PREFECTCORE

PREFECTCLOUD

SYNOPSIS

Introduction

The Nationals' engineering team uses data to analyze players and games, optimizing play metrics and high-level decisions.

Challenge

Employ a data engineering framework to reliably process critical statistics for data analysts to deliver strategic recommendations on scouting, positioning, and training regiments, in a timely manner.

Solution

Incrementally adopt Prefect for data pipelines to standardize, modernize development practices, and empower baseball data analysts by granting them control over data pipelines with Prefect Cloud's intuitive UI.

Results

Comprehensive deployment of daily jobs to promptly deliver data to analysts and standardize their workflows to reliably deliver critical recommendations to coaches and executives based on performance metrics.

Introduction

While the baseball players of the Washington Nationals train for the season by becoming stronger and better than their opponents, the Nationals' data engineering team trains data models to make their organization smarter than all others—using data-driven insights to win more baseball games. This pursuit entails aggregating data from a variety of sources, normalizing freshly-minted data with historical trends, and staying informed of engineering advances to gain every attainable edge. Their insights provide recommendations for in-game strategy, advanced scouting, roster construction, and player development.

Challenge

The use cases for data tooling are plentiful for the Nationals' data team and span a variety of opportunities for analysis. By capturing data from several APIs, engineers can funnel post-game data for in-game strategy recommendations like defensive positioning or the ideal pitch type and pitch location to a particular batter. In the off-season, recruiters use data to drive advanced scouting, while analysts are tasked with roster construction to determine a player's value in congruency to the existing team. Even after a player's recruitment, data drives their improvement regiment by targeting shortcomings and enabling data-driven adjustments to the roster.

Given the lifecycle of data from high-level recruitment decisions to granular player performance metrics, the importance of timely, high-quality data is paramount. Importing data from a wide range of sources like radar guns to various APIs like MLB Stats, consists of considerable challenges in data normalization and synthesis. Additionally, the team's processing requires reliability and regularity with the constant influx of new data which must be synchronized for analysis. If a stage in data processing breaks down, then data cannot be prepared for post-game analyses and the needed recommendations for coaches are not available in time for the next game. These minute engineering details cascade onto the field, all for the purpose of winning more baseball games by increasingly larger margins.

> “ *By reliably importing the data several times a day, our analysts can then make use of that data by performing analyses, writing reports, and sharing those results with our front office, coaching staff, and players.*
>
> LEE MENDELOWITZ

As the future becomes more data-driven and the current landscape of data capture tooling continues to grow, a data system with the stability to scale is absolutely necessary. Currently, the primary means of capturing in-game data is radar-based, but recent initiatives have pushed for an optical sensor-based system at the professional level and baseball teams now use data scientists to provide statistics. These systems require scalable systems to decipher and translate data for coaches and managers to make game-time adjustments.

Solution

The Nationals' data team required a flexible framework to reliably handle their data imports and accommodate their technical needs. They needed a tool capable of pulling from multiple APIs, including Rapsodo radar data capture and MLB Stats, on a regular schedule with verbose monitoring into the status of their data pipelines, to be able to easily detect when jobs fail. For the team, the right tool could potentially reduce the amount of time needed to investigate failures and utilize conditional logic to automate recovery scenarios in case of failure. By automating these processes, the team could devote more time to business-critical objectives: improving the performance of their players and setting a course for team development.

Though still in the early stages of development, Prefect Core's workflow semantics, combined with Prefect's Cloud orchestration components, seemed like a promising tool for their needs. In particular, the tool could be integrated into their existing system consisting of Rundeck jobs, so engineers could capitalize on Prefect Core's functionality, schedule to run with Rundeck, until their incremental adoption enabled a full transition to Prefect Cloud. The team could continue to operate with legacy systems, gradually integrate Prefect to grow with the tool, and define subsequent pipelines with Prefect using new data imports. This incremental adoption allowed the team to transition gradually without compromising business critical workflows.

> “ *Prefect Core provides a really nice, clean set of easy to use features for constructing data pipelines (Tasks, Flows, States, Results) and it's easy to model the dependencies between tasks. There was a very short learning curve for getting started, and the framework provided by Prefect core was already way better than our crude implementation of try/catch statements.*
>
> LEE MENDELOWITZ

Prefect's offerings provided the right set of solutions for the team —Prefect Core's clearly defined components provided a framework for their system and Prefect Cloud served as their command center. The ingredients for semantic pipeline tools were clearly defined and provided a basis for the team to achieve consonance across their distinctive data jobs. These jobs could vary considerably, but Cloud served as a platform to centralize monitoring and orchestration of many disparate systems across environments. Additionally, newly engineered pipelines could be easily deployed into containerized production and registered to Cloud under version control, enabling a robust development lifecycle and ensuring fresh data always finds its way to analysts.

> " *It's really nice to be able to define the Flows precisely, using code that's under version control. Features like Tasks, Task dependencies, Task retries, mapping make it easy to write robust data imports and data pipelines.*
>
> LEE MENDELOWITZ

As the adoption of Prefect has increased internally with new pipelines, analysts are empowered by Prefect's functionality without deep software engineering knowledge. Analysts can freely navigate Prefect Cloud's intuitive user interface to watch their data populate in real-time. Even without a deep comprehension of the underlying containerized environments or infrastructure, a data analyst who specializes on pitcher analysis is permissioned to run impromptu data pipelines and can escalate to backend engineers if issues arise. If a pipeline that runs data imports fails at some point processing a large set of games, these API errors can be seen clearly in the Cloud Logs and rectified quickly, to allow for analysts to maintain focus on analyzing pitch type, plate crossing, and how much a pitch broke vertically and horizontally. Ultimately, the efforts of the small-scale data team to productionize a much larger number team of analysts was successful, empowering analysts to trigger ad-hoc parameterized Flows and insight into their scheduled pipelines.

Results

Reliable data imports from multiple sources can be scheduled multiple times a day for various teams across the organization.

Reduction in negative engineering, allowing analysts to focus on applying their specialty rather than exhaustive pipeline issues.

Prefect Cloud serves as a central data platform to monitor and orchestrate data imports of many disparate systems across environments.

Prefect's accessible APIs allowed for incremental adoption into existing workflows.

Despite production workflows executing with RunDeck, the data team was able to slowly adopt the Prefect module and workflow semantics without disrupting business critical pipelines.

> " *We have been able stay on top of the data flows we've moved to Prefect easily. Seeing failures, successes, and outages in a timely and clear fashion has let us inform stakeholders what's up with the data flows.*
>
> CHRIS JORDAN

Accelerated deployment of new jobs to Prefect Cloud without security concerns.

Prefect Core provides workflow semantics for repeatable data pipeline development, with modular pieces that are easily interpretable with Prefect Cloud's UI.

Cloud UI provides out-of-the-box monitoring and eliminates security concerns due to Prefect's Hybrid Execution Model.

> " *I especially appreciate the wrappers for logging and notification. I've had to implement these things on my own before, and having a tool to just do it saved a lot of time.*
>
> CHRIS JORDAN

Developing a positive feedback loop between data capture, reporting, and analysis.

The accelerated production of data pipelines and the granular monitoring provides the infrastructure for the entire organization's data-driven decision making.

> " *...any team that has data pipelines that run daily in the cloud could benefit from using Prefect Core and Prefect Cloud, no matter the industry.*
>
> LEE MENDELOWITZ