David Isaac Belais

Portland OR | 503-267-0942 | david@belais.me | david.belais.me | github.com/davebelais

Summary

I am a highly productive software and data engineer with 18 years of relevant experience.

I pride myself in:

- Creating resilient, maintainable, integrous data products
- Authoring elegant, bulletproof, type-annotated, well-formed, thoroughly tested, well documented, distributable Python libraries, CLIs, web APIs, SDKs, and Spark jobs
- Writing *readable* and efficient SQL
- Designing efficient, maintainable, testable, continuously integrated and deployed, modern software systems
- Planning development work with clarity, flexibility, parallel execution, and collaboration in mind
- Leading engineering teams with complex and ambiguous directives towards clear, executable road maps
- Condensing fact from the vapor of nuance

Skills

I have professional experience with (not exhaustive):

- Platforms: Databricks, Snowflake, Amazon Web Services (AWS - including Lambda, EMR, Aurora, IAM, Cloudformation, EC2, S3)
- Languages: Python, SQL, Javascript, C++, HTML, XML, PHP, WSDL, Rust
- Databases and query engines: Databricks Lakehouse, Delta Lake, Snowflake, Terradata, Netezza, Hive, Presto, DuckDB, PostgreSQL, MySQL, SQL Server, Oracle, IBM DB2, SQLite, MariaDB
- Applications, Services and Frameworks: Apache Spark, Apache Kafka, SQLAlchemy, FastAPI, Flask, Docker, Terraform, Linux, Unix, Github Actions, Jenkins, Kubernetes, Hadoop, Copilot
- Protocols and Specifications: Open API (Swagger), SOAP, MIME, AS2 (for GDSN data pools), ASGI, WSGI
- Distributed File Systems: DBFS, S3, HDFS

Experience

Nike | Lead Data Engineer - Sustainability Analytics | March 2021 - June 2025

- I lead and mentored a team of, variably, 4-8 data/software engineers in developing data and software products supporting analysts, data scientists, environmental scientists, product developers, and sustainability professionals in assessing and mitigating Nike's environmental impacts.
- I implemented ELT and ETL data pipelines leveraging Databricks Delta Lake using Python, PySpark, Spark SQL, Snowflake SQL, and Amazon EMR (Python, PySpark, Spark SQL, Hadoop/HDFS and Hive/HQL)—employing patterns using batch, micro-batch, streaming (Apache Kafka and Spark) and Delta live tables, CDC (change data capture) from Oracle and PostgreSQL, and pub/sub from Amazon SQS and SNS. Overall, my redesign of Sustainability Analytics' data pipelines reduced compute costs by 80% as compared with equivalent legacy pipelines.
- I authored Python web APIs using FastAPI and SQLAlchemy on AWS Lambda, using Okta OAuth2 authentication, Aurora PostgreSQL for persistence, Route 53 and Amazon API Gateway to route requests, deployed using Terraform for infrastructure-as-code. This web API facilitated preemptive mitigation of environmental impacts by facilitating pre-manufacture scenario modeling in client product development systems.
- I designed and built complete systems for calculating material and product footprints comprised of individually testable component python libraries, permitting us to fully employ test-driven development, and thereby safely make use of continuous integration and deployment (CI/CD) with Jenkins and Github Actions, and permitting us to often release multiple features daily.
- I employed dimensional modeling and type 2 slowly changing dimensions in our Databricks Delta Lake, Snowflake databases, and (prior to 2023) AWS EMR S3/hive data lake in order to address obstacles to replicating historically reported metrics which are required for regulatory audits.
- I authored foundational data products for our Environmental Health & Safety data ingested from our 3rd-party EHS reporting system (Enablon) via their OData API into our Databricks Delta Lake, Snowflake, and (prior to 2023) AWS EMR + S3 + hive/presto data lake.
- I authored enterprise developer tools including python CLIs (command line interfaces), libraries, and SDKs (internal and 3rd party) for CI/CD job deployment and orchestration on Databricks and Airflow, for data validation, generating data model diagrams, schema versioning and migration, and extending SQLAlchemy's ORM for simultaneous multi-dialect support and view management supporting OLAP databases including Databricks Delta Lake, Snowflake, and Hive and extending view management functionality to OLTP databases including PostgreSQL and SQLite.

BICP @ Nike | Lead Data Engineer - Sustainability Analytics | March 2020 - March 2021

- I lead and mentored a small team of 3-4 data engineers in building foundational data products supporting sustainability initiatives
- I developed a Python library augmenting the SQLAlchemy dialects for Snowflake SQL, Databricks SQL, and Hive/HQL to support full ORM (object relational mapping) functionality, and to add view management functionality for these dialects as well as PostgreSQL and SQLite, in order to facilitate fully aligned multi-platform publication of Nike data products for all databases/data lakes supported by Nike Data and Analytics orgs. This facilitated seamless deployment and validation of data products accessible on multiple platforms.
- I authored a framework (Python library) facilitating deployment of Nike ELT/ETL Spark jobs seamlessly as either an Apache Airflow DAG, Spark on AWS EMR job, or Spark on Databricks job, allowing us to use the most cost effective solution for a given scenario, and saving upwards of 90% on compute costs for small jobs and reducing compute time for several large jobs (while moderately reducing cost) from hours to minutes.
- I authored a Python library abstracting and applying a common interface (as well as aligning/adding support for date partitions and check-pointing) to the various file systems leveraged by Nike including S3, DBFS, Hadoop/HDFS, local/EBS, and Box. This foresight dramatically expedited subsequent platform migrations from our S3/hive data lake and Snowflake databases to Databricks Delta Lake.

BICP @ Nike | Senior Data Engineer - Sustainability Analytics | January 2020 - March 2020

- I created dimensional models for Sustainability Analytics' Snowflake data warehouse and Amazon EMR Hive/S3 data lake
- I planned a Jira backlog/roadmap to integrate shadow tech developed as part of a contract statement of work into data products and pipelines aligned with Nike's enterprise data strategy and governance, and supportable long-term by the Nike Global Technology organization
- I established Python + SQL exams for screening new hires to fill out the additional engineering roles needed for the team

The Kroger Co. | Lead Data Engineer - Web & Digital Analytics | May 2018 - November 2019

- I authored data products using Apache Spark (PySpark, Spark SQL) marrying fact, dimensional and taxonomy data we ingested from click-streams using Apache Kafka, enterprise system busses (in-store sales data) using AS2/MIME (sync and async), NoSQL databases including Apache Cassandra and Mongo DB, and produced data productspublished to transactional and analytics databases and data lakes including IBM DB2, Oracle, Netezza, Cloudera, Hadoop/HDFS Hive/Presto/Impala, and SQL Server. The data products we produced correlating in-store and digital sales, EBIDTA, sell-through, and pricing/promotions contributed to decisions resulting in a 56% increase in e-commerce sales in 2018 vs 2017, and a 67% increase in e-commerce sales in 2019 vs 2018.
- I developed integrations synchronizing in-store prices and promotions from our ERP with Magento Commerce, developing a meta-programming library for auto-generating Python SDKs based on that platform's Open API (Swagger) schema which would later serve as the conceptual seed for the `oapi` Python library.
- I authored pricing feeds using Python, SQLAlchemy, and IBM DB2 resolving type 2 slowly changing dimensions from enterprise pricing systems for consumption by IBM Websphere Commerce (by way of Informatica), synchronizing online with in-store pricing for ~140 disparate sets of prices for 500,000+ SKUs frequently involving complex and layered promotions and application of manufacturer and store coupons.

The Kroger Co. | Lead Data Engineer - Product Information Management | November 2013 - May 2018

- I lead development of multi-platform transactional and analytics data products consolidating and normalizing dimensional, fact, and click-stream data from disparate subsidiary and partner systems' transactional databases, streaming platforms, web APIs, mainframes and system busses using Spark, Cloudera, Hadoop/HDFS, Hive/HQL, Presto, Netezza, IBM DB2, Python, SQL Server, SQLAlchemy, Apache Cassandra, Mongo DB, and AS2.
- I engineered algorithms for scoring semi-structured data, and performing human-in-the loop data validation and auditing for product descriptions, specifications and photography acquired through trading partners using Python, Flask, SQLalchemy, SQL Server, IBM DB2
- I established source management capabilities for identifying sourcing and retrieval mechanisms for ingesting augmented product information for digital sales channels using Python + Flask + SQLAlchemy + SQL Server to manage complex distributor/vendor/manufacturer relationships — integrating with multiple internal ERP systems, IBM DB2 and Oracle data warehouses, GS1 SOAP/WSDL/XML web APIs, and XML documents from GDSN data pools ingested via AS2.
- I worked with our team of digital content creators to capture and assess productivity metrics for authoring of product page and search attribution content, in order to facilitate nuanced accountability and identify replicable patterns employed by the most prolific authors. For this initiative I used Python, Pandas, SQL Server, Flask, and SQLAlchemy. These efforts resulted in a throughput increase per/author of greater than 100% every year, and in 2018-2019 the teams I supported boasted overall per/associate output greater than 7 times that of competing outsourced initiatives, while maintaining 30% lower error rates as determined by internal audits routinely performed by buyers and category managers.
- I developed an application facilitating internal review of digital content for e-commerce product pages' accuracy and style utilized by stakeholders (buyers, planners, and category managers) using a stack including Python, Flask, SQLAlchemy, SQL Server, HTML, CSS, and Javascript.

Compucom @ The Kroger Co. (Fred Meyer Stores Inc.) | Business Systems Analyst - Ecommerce | March 2011 - November 2013

- I researched, designed, and prototyped Fred Meyer's (and later Kroger's) product information management system for customer-facing digital initiatives.
- I collaborated with Fred Meyer's technology partner, 1WorldSync, to establish a roadmap, data model, and procedures for sourcing and validating product data from GDSN data pools for use in digital sales channels through SOAP/WSDL APIs and XML MIME/AS2 synchronous and asynchronous messaging protocols, landing the data in Kroger's IBM DB2, Oracle, and SQL Server databases.
- I extracted and cleansed data from GDSN data pools using Python and lxml, generating HTML/CSS/Javascript intranet pages and HTML emails for buyer review, and collected feedback using forms leveraging Python, SQLAlchemy, WSGI/Flask and SQL Server.

Dissent Graphics Inc. | Full-Stack Developer | January 2008 - March 2011

- I designed and developed web applications for clients including: The Garrigan Lyman Group, Microsoft, Best Buy, Avenue A Razorfish, Nereus Communications, BlackEyedPeas.com, TeeFury.com, the Travel Channel's Man v. Food, TheWho.com, Custom Rights, Hello Minor, ExoticTravelers.com, and the ACLU of Oregon using Python, Django, PHP, SQL (MySQL and PostgreSQL), Javascript, and Actionscript.

Education

- Portland State University | Computer Science (Postbaccalaureate) | 2018
- The Art Institute of Portland | Conmputer Generated Imaging, Visual Effect & Animation | Bachelor of Science | 2007
- Portland State University | Web Design | 2002 - 2004
- Loyola Marymount University | Fine Arts | 2000 - 2001

Open Source Projects

...because code examples are worth a thousand interview questions!

- git-author-stats: A CLI and library for extracting periodic author "stats" (insertions and deletions) for a Git repository or Github organization
- dependence: A CLI and library for aligning a python projects' declared dependencies with the package versions installed in the environment in which dependence is executed, and for "freezing" recursively resolved package dependencies (like pip freeze, but for a package, instead of the entire environment).
- maya-zen-tools: An Autodesk Maya extension providing modeling tools for manipulating a polygon mesh using dynamically created NURBS curves and surfaces to distribute vertices and/or UVs
- oapi: A python library for generating client SDKs from Open API documents
- gittable: A CLI and library for performing common, but complex, development and CI/CD tasks for a Git repository, such as tagging a commit with your current project/package version and downloading or accessing specific file(s) from a remote repository (including non-public repos)

Please see github.com/davebelais and github.com/enorganic for additional code examples.

Certifications

- JQL for Admins
- Jira Automation for Admins
- AWS Certified Big Data - Specialty
- AWS Certified Cloud Practitioner