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/Software Engineer - Sustainability Analytics | March 2021 - June 2025

Platforms: Databricks, Snowflake, Amazon Web Services (AWS - including Lambda, EMR, Aurora, IAM, Cloudformation, EC2, S3)
Languages: Python, SQL, Javascript
Databases and query engines: Databricks Lakehouse, Delta lake, Snowflake, Terradata, Hive, Presto, PostgreSQL, Oracle, SQLite
Applications, Services and Frameworks: Apache Spark, PySpark, Apache Kafka, SQLAlchemy, Alembic, FastAPI, Terraform, Docker, Linux, Github Actions, Jenkins, Hadoop, Copilot
Protocols and Specifications: Open API (Swagger), ASGI
Distributed File Systems: DBFS, S3, Hadoop/HDFS **CI/CD:** Jenkins, Github Actions

- 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, and HQL)—employing patterns using batch, micro-batch, streaming (Apache Kafka and Spark) and Delta live tables, reducing 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, and 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/Software Engineer - Sustainability Analytics | March 2020 - March 2021

Platforms: Snowflake, Amazon Web Services (AWS - including Lambda, EMR, Aurora, IAM, Cloudformation, EC2, S3)
Languages: Python, SQL, Rust
Databases and query engines: Snowflake, Terradata, Hive, Presto, PostgreSQL, Oracle, SQLite
Applications, Services and Frameworks: Apache Spark, SQLAlchemy, Alembic, Docker, Linux, Terraform, Github Actions, Jenkins, Hadoop
Protocols and Specifications: Open API (Swagger)
Distributed File Systems: S3, Hadoop/HDFS
CI/CD: Jenkins, Github Actions

- 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/Software Engineer - Sustainability Analytics | January 2020 - March 2020

Platforms: Snowflake, Amazon Web Services (AWS - including Lambda, EMR, Aurora, IAM, Cloudformation, EC2, S3)
Languages: Python, SQL, Rust
Databases and query engines: Snowflake, Terradata, Hive, Presto, PostgreSQL, Oracle, SQLite
Applications, Services and Frameworks: Apache Spark, SQLAlchemy, Alembic, Docker, Linux, Jenkins, Hadoop
Protocols and Specifications: Open API (Swagger), ASGI
Distributed File Systems: S3, HDFS
Infrastructure as Code: Terraform
CI/CD: Jenkins

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

Languages: Python, SQL, Javascript, HTML, XML, WSDL
Databases and query engines: Netezza, Hive, Presto, SQL Server, IBM DB2, SQLite, Apache Cassandra, Mongo DB
Applications, Services and Frameworks: SQLAlchemy, Flask, Hadoop, Magento Commerce, IBM Websphere Commerce
Protocols and Specifications: Open API (Swagger), SOAP, MIME, AS2 (for GDSN data pools), WSGI
Distributed File Systems: Hadoop/HDFS

- I lead development of data products marrying fact, dimensional and taxonomy data from relational and NoSQL databases including Apache Cassandra, Mongo DB, IBM DB2, Oracle, and SQL Server with sales and clickstream data from Hadoop/HDFS on Cloudera, exposing analytics to buyers and planners correlating digital and store sales and EBITDA with inventory,sell-through, prices, and promotional events—contributing to decisions resulting in a 56% increase in e-commerce sales in 2018 vs 2017, and a 67% increase in ecommerce sales in 2019 vs 2018.
- Pricing/promotions and product information integration services for Magento Commerce.

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

Languages: Python, SQL, Javascript, HTML, XML, WSDL
Databases and query engines: Netezza, Hive, Presto, SQL Server, IBM DB2, SQLite, Apache Cassandra, Mongo DB
Applications, Services and Frameworks: SQLAlchemy, Flask, Hadoop
Protocols and Specifications: SOAP, MIME, AS2 (for GDSN data pools), WSGI

- I lead development of multi-platform (Spark/Hive/Presto, Netezza, DB2, Python, SQL Server, SQLAlchemy) OLAP and OLTP data products to ingest, consolidate and normalize sales, dimensional, and click-stream data from disparate subsidiary and partner systems' transactional databases, streaming platforms, APIs, and mainframes.
- 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 (Python, SQL).
- I established source-management capabilities for inbound data to handle complex retailer/vendor/manufacturer relationships (Python, SQL).
- Collaborated with emerging digital initiatives to ensure the capture of all metrics needed to facilitate accountability and continuous operational improvement.

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

Languages: Python, SQL, Javascript, HTML, XML, WSDL
OLTP Databases: SQL Server, IBM DB2, SQLite
Applications, Services and Frameworks: SQLAlchemy
Protocols and Specifications: SOAP, MIME, AS2 (for GDSN data pools)

- 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.

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

Languages: Python, PHP, SQL, Javascript, Actionscript

- 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.

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