

Ibis: Easy, Performant, and Portable Python API for Data Analytics

Thank you for downloading this Voltron Data resource. Carahsoft is the distributor for Voltron Data's AI and ML solutions available via NASA SEWP V, ITES-SW2, NASPO ValuePoint, and many more contract vehicles.

To learn how to take the next step toward acquiring Voltron Data's solutions, please check out the following resources and information:



For Voltron Data overview:
carah.io/Voltron-Data



For upcoming events:
carah.io/Voltron-Events



For additional resources:
carah.io/Voltron-Resources



For additional Artificial Intelligence:
carah.io/ai-solutions



To set up a meeting:
VoltronData@carahsoft.com



To purchase, check out the contract vehicles available for procurement:
carah.io/procurement

Feb 28, 2023

Ibis: Easy, Performant, and Portable Python API for Data Analytics

Fernanda Foertter



Imagine this: Some VPs at Big Mart Corp get together and create a fantastic idea that could potentially save millions of dollars. What if we used data from Purchasing, Sales, and Warehouse to improve the efficiency of moving goods and time the purchase of raw materials? Sounds great, but there's just one problem: to analyze this complicated dance between space, inventory, and sales cycles, it needs to connect to multiple data sources. You're just the person they asked to get it done.



What if you could write pandas-like analytics and run it at the source?

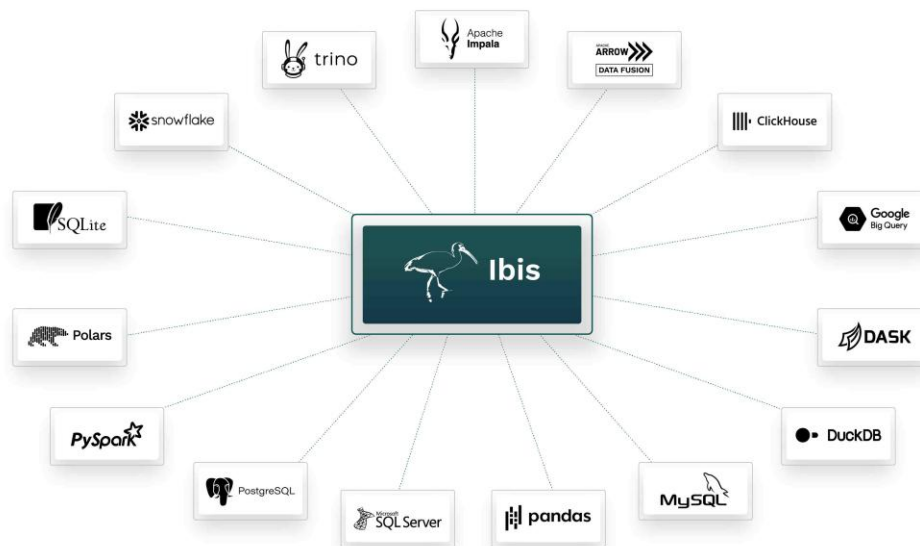
With Ibis, you can!

In the current world, a team of experts from each department would be assembled, they would meet to decide what data is needed, and how to do it. But each source connects in a different way, each department may even house data in multiple locations. To amortize the impact on business and their respective teams, you'll likely accomplish this in phases. In the end, the solution would involve some data duplication, monolithic downloads, and

loads of CSV files. Not to mention maintaining code written specifically for that Hadoop, SQL, or Spark datasource, maybe even in different languages. As complexity grows it becomes harder to maintain, a major team member wins the lottery and leaves and wrote everything in Rust and no one left behind knows how to keep this going. Sound familiar? Well, there's another way with [Ibis](#).

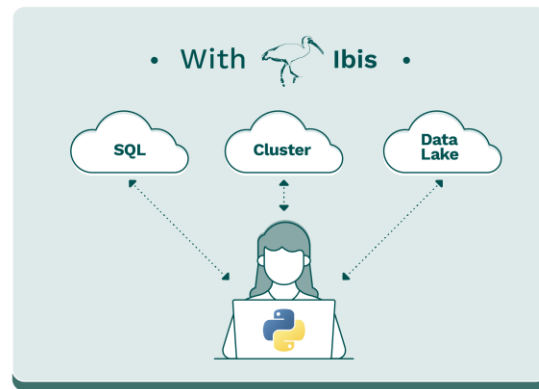
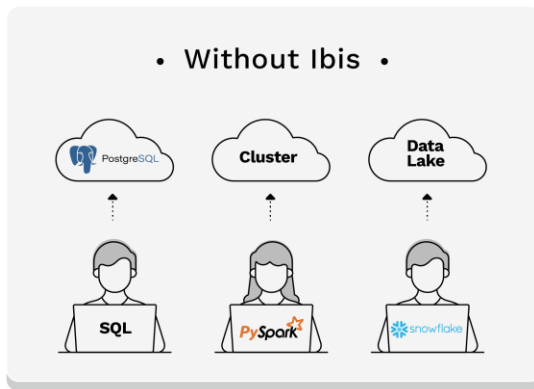
All You Need is One

Because Ibis compiles and generates code for many backends, the only language needed is Python. This makes maintaining the codebase very easy. [Ibis currently supports 15 backends and counting!](#)



Reduce Technical Debt

Using a single language simplifies things, but Ibis is also growing and evolving, adding more backends every day. Instead of learning different interfaces or making sure packages don't conflict with one another, [Ibis makes application maintenance and development a breeze.](#)



Fly Like a Bird and Leave the Bear Behind.

Queries shipped by Ibis run at the source. This means typical analytics can be run far away at the source, and only the results are shipped back to the application, in memory. No need to bring everything to pandas either, but [still enjoy the pandas-like syntax](#). And Ibis performs just as well as queries written by hand for the native database engine.

```
#pandas
df['col1']
#ibis
table_expr['col1']
```

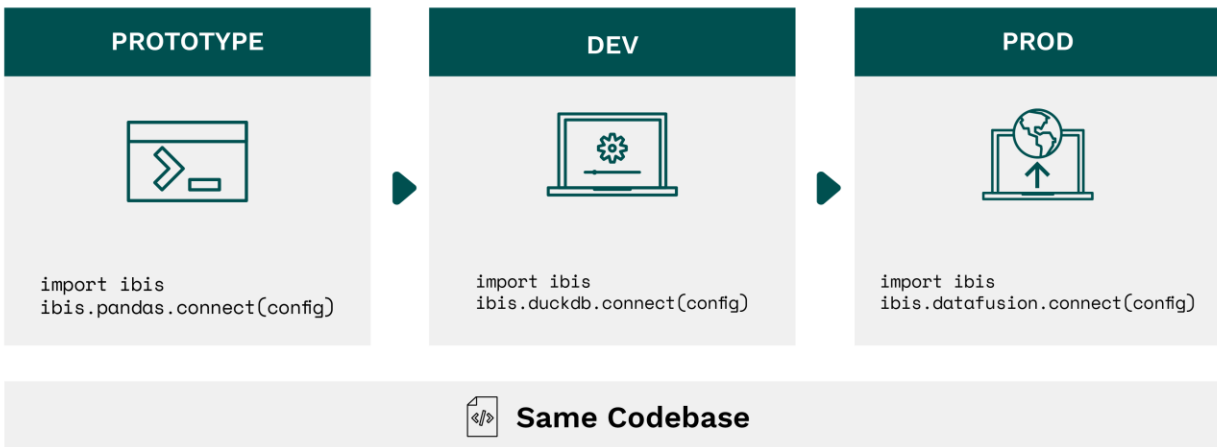
PYTHON

```
#pandas
df[['col', 'val']].groupby(['col']).sum()
#ibis
from ibis import _
#the underscore (_) API allows us to reference parent expressions
table_expr.groupby(['col']).aggregate(_['val'].sum())
```

PYTHON

From Prototype, to Dev, to Prod with a Simple Switch

Start coding right away and use pandas dataframes as a backend, fill it with dummy data to prototype the work. When ready, move onto a dev environment where perhaps some simple postgres databases were created for testing. When ready, deploy to prod and connect to the big SQL database. All of these steps only require a backend change and the rest of your code remains the same. Yes, it's that Incredible.



Expressive and Extensible

Ibis provides a user-facing API that allows users to define operations. For example, create [reductions](#) or translate string dates to Julian. With Ibis, you can add new operations, optimizations, and custom APIs.

```
import ibis.expr.datatypes as dt
import ibis.expr.rules as rlz
from ibis.expr.operations import ValueOp

class JulianDay(ValueOp):
    arg = rlz.string

    output_dtype = dt.float32
    output_shape = rlz.shape_like('arg')
```

PYTHON

Join the Community

[Voltron Data](#) is a [major contributor to Ibis](#), which was developed by one of our co-founders and CTO, Wes McKinney. Join us to help make Ibis the future of easy, performant, and portable data analytics.