

# Ask the Expert

Coming Soon: SAS<sup>®</sup> Viya<sup>®</sup> Workbench!

Joe Madden, Senior Product Manager





# Joe Madden

Senior Product Manager

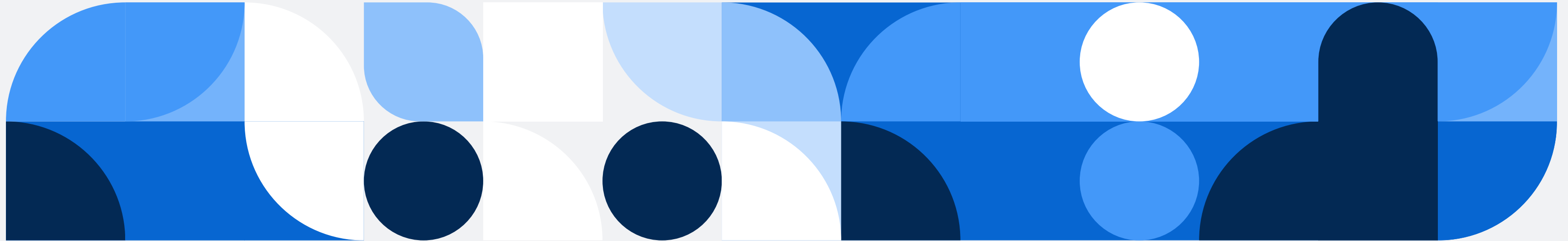
Joe Madden works in the Analytics and AI Product Strategy division at SAS. He joined SAS in 2022 and primarily focuses on enabling customers with open source solutions. Joe received an MBA from Boston University's Questrom School of Business in 2017.

# SAS Viya Workbench

## Product Overview

January 2024





# SAS Viya Workbench

A data science canvas  
for modelers,  
developers and  
programmers

- Self-service and flexible
- On-demand GPU/CPU compute power
- AWS-only in first release
- Open-source integration through Visual Studio Code or Jupyter Notebook
- High performance analytics using the latest SAS procedures and APIs
- Develop analytical and machine learning models using SAS code, Python and R

# Data Science Development Challenges

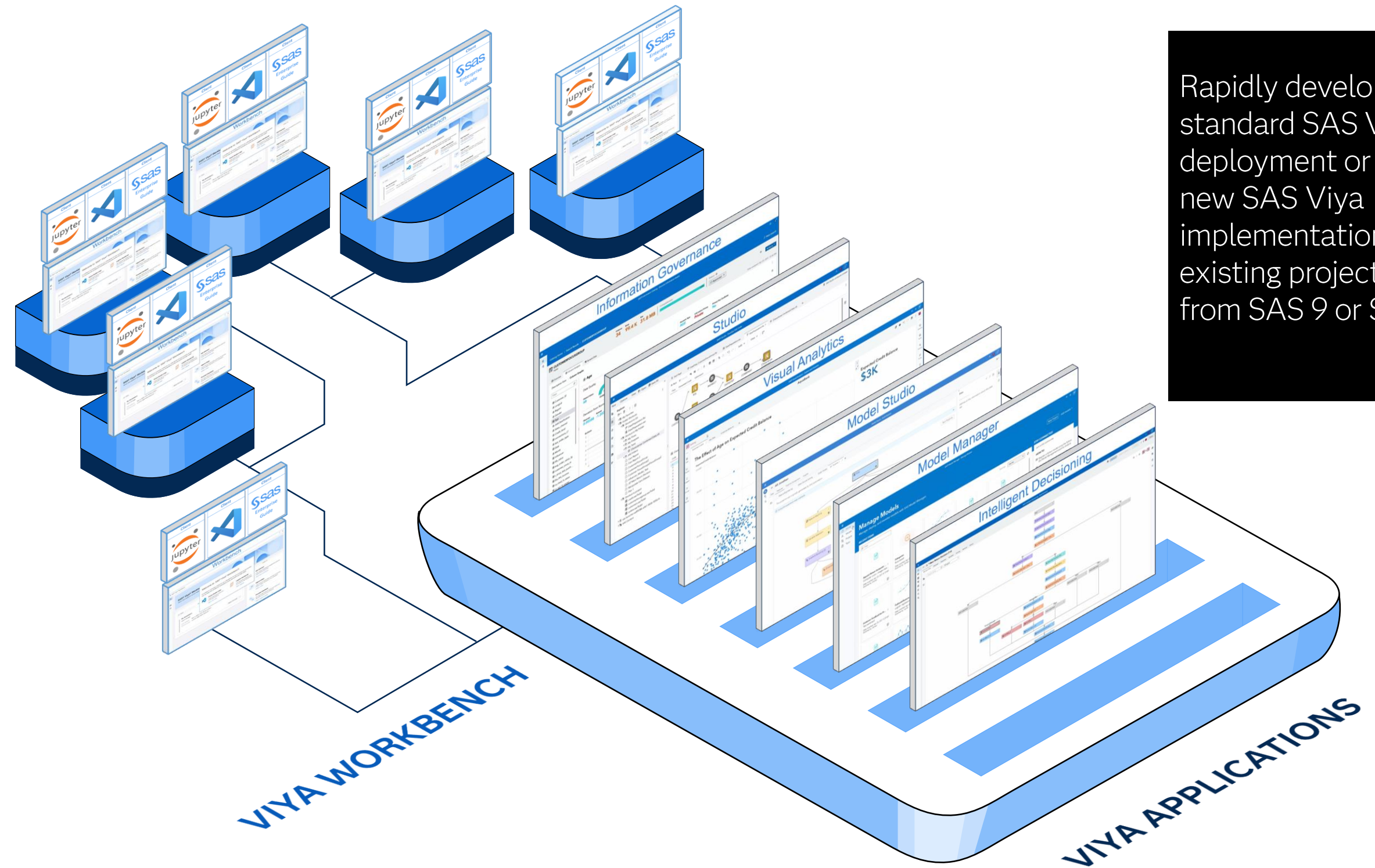
## Systems, People and Processes

- Finding and validating data
- Collaborating across groups, programming languages and tools
- Provisioning the right-sized environment without Administrator requirements
- Reusing existing work without the need to modify
- Scaling and transferring work to an enterprise platform
- Reducing cost and complexity of the data science workspace
- Balancing data security and interoperability

# What is SAS Viya Workbench?

A data science and analytics canvas to develop, experiment and collaborate

Self-provisioned via a lightweight, scalable deployment environment – on demand

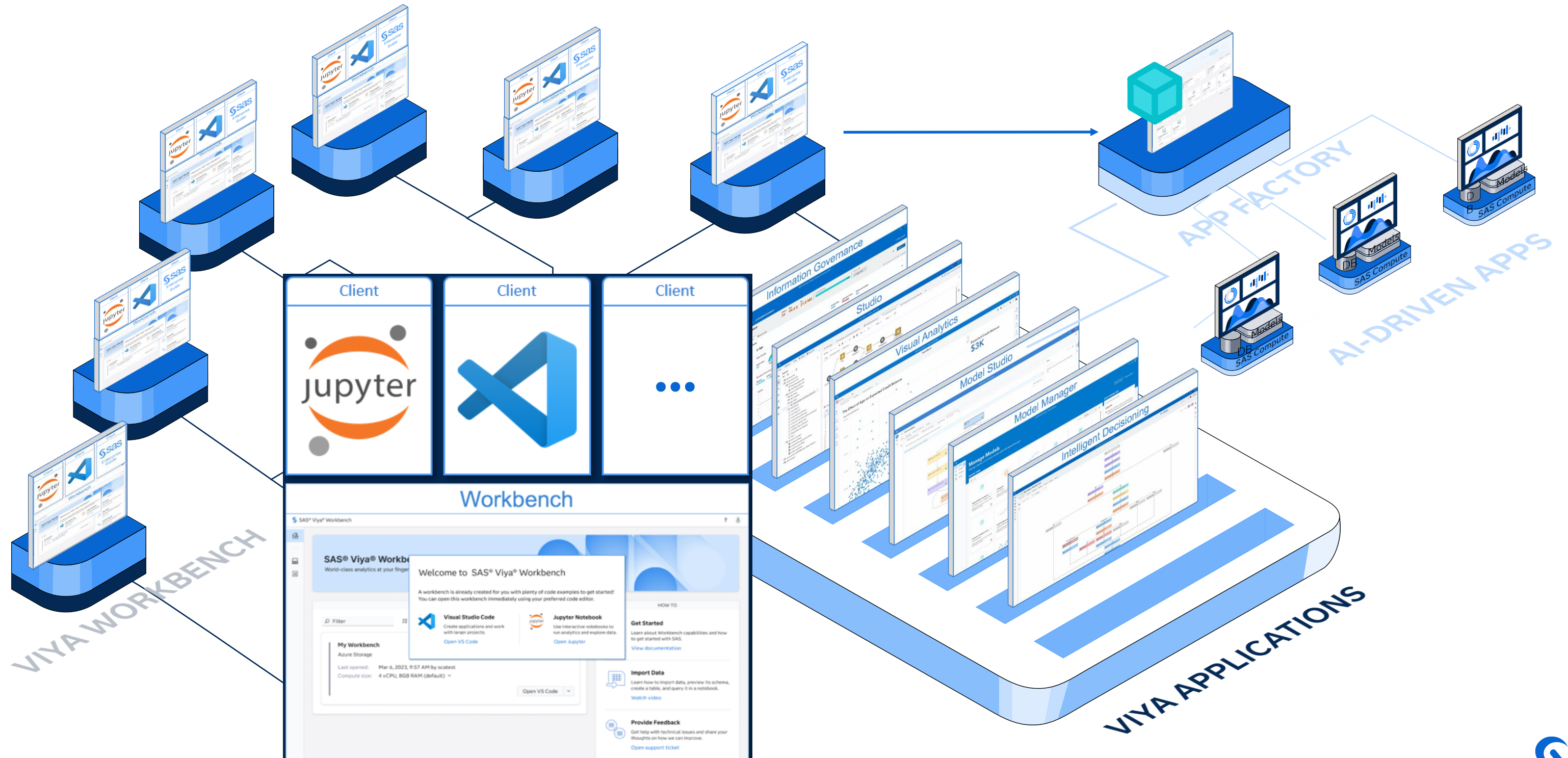


Rapidly develop alongside a standard SAS Viya deployment or share into a new SAS Viya implementation. Use existing projects or code from SAS 9 or SAS Viya



# SAS Viya Workbench in the SAS Ecosystem

Create in a variety of languages including SAS, Python or R and use existing code from SAS 9 or SAS Viya projects



# SAS Viya Workbench

## Key Features



### Data Science Canvas

An easy to use, lightweight and scalable data science modeling and analytics workspace



### Best in Class Analytics

Latest SAS Analytical Procedures (PROC) and APIs



### Open-Source Coding

VSCoDe and Jupyter Notebook interfaces. SAS and Python compute runtimes



### Cloud-Native

Elastic cloud compute, on AWS



### Efficient Compute Engine

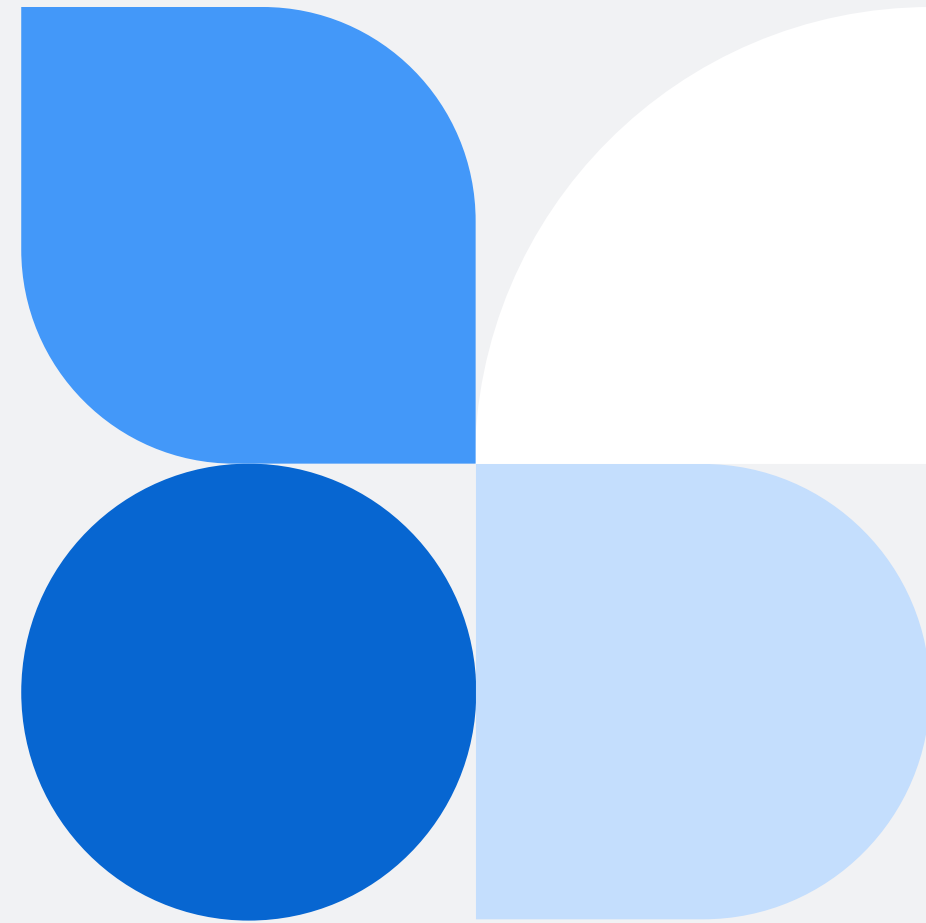
Self-service provisioning and termination with scalable GPU to meet workload



### Governance and Security

Managed deployment with update and backup capabilities. Industry specific security compliance





# Benefits of SAS Viya Workbench for Programmers, Data Scientists and Developers

- **Easily Deployed Data Science Workspace**  
A canvas to conduct exploratory data analysis and develop analytical and machine learning models
- **Latest analytical procedures and APIs**  
Access the latest SAS PROCs and APIs with the ability to update and backup projects including GIT version control
- **Choice of coding languages**  
Use SAS, Python or R through Visual Studio Code or Jupyter Notebook
- **Team Collaboration**  
Share models, analytics projects and experiments
- **Flexible and Scalable**  
Scalable computing with on-demand server size to match workload requirements – all with fast run times

# Quick Tour of SAS Viya Workbench

Real-world example for Data Science teams

## SAS Viya Workbench

Chris Parrish, Sr. Data Scientist  
Financial Services  
January 2024



Copyright © SAS Institute Inc. All rights reserved.

# Run SAS Code from SAS 9 or SAS Viya

The screenshot displays the SAS Viya interface. On the left, a code editor shows SAS code for a regression model. The code includes comments, a title, and PROC IMPORT statements for training and test data. It also includes PROC PRINT and PROC FORMAT statements. On the right, two data tables are displayed. The first table, titled 'Build SAS regression model to predict bike sharing demand Original data from bike\_sharing\_demand\_train.csv', shows the first five rows of the original data. The second table, titled 'Build SAS regression model to predict bike sharing demand Updated bike\_train data with formatted values and new date and time columns', shows the first four rows of the updated data, which includes formatted values for season, holiday, workingday, weather, temp, atemp, humidity, windspeed, casual, and registered.

```
1 *****
2 EXAMPLE: linear_bike.sas
3 DATA: bike_sharing_demand_train, bike_sharing_demand_test
4 DESCRIPTION: This dataset contains the hourly and daily count of rental bikes
5 with the corresponding weather and seasonal information.
6 PURPOSE: This example shows how to analyze data with visualization tools
7 such as SGLOT and how to fit a regression model to predict bike
8 demand using the REGSELECT procedure.
9
10 *****/
11 title 'Build SAS regression model to predict bike sharing demand';
12
13
14 /*****
15 Load the training and test data.
16 *****/
17 proc import
18   datafile="GWORKSPACE_PATH./data/bike_sharing_demand_train.csv"
19   out=bike_train_origin
20   dbms=csv
21   replace;
22 run;
23 proc import
24   datafile="GWORKSPACE_PATH./data/bike_sharing_demand_test.csv"
25   out=bike_test_origin
26   dbms=csv
27   replace;
28 run;
29
30 /*****
31 Print a few rows to show the original data.
32 *****/
33 title2 'Original data from bike_sharing_demand_train.csv';
34 proc print data=bike_train_origin (obs=5); run;
35
36 /*****
37 The "datetime" column contains too many levels for meaningful analysis.
38 We will use SAS functions to split the "datetime" column into:
39 "date", "hour", "weekday" (day of the week) and "month".
40
41 We will also use FORMAT to format the "season" and "weather" columns to make
42 them more readable to users. When calling SAS analytic procedures, the
43 formatted values can also be used instead of the raw values.
44 *****/
45 proc format;
```

Obs	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed
1	01JAN11:00:00:00	1	0	0	1	9.84	14.395	81	0
2	01JAN11:01:00:00	1	0	0	1	9.02	13.635	80	0
3	01JAN11:02:00:00	1	0	0	1	9.02	13.635	80	0
4	01JAN11:03:00:00	1	0	0	1	9.84	14.395	75	0
5	01JAN11:04:00:00	1	0	0	1	9.84	14.395	75	0

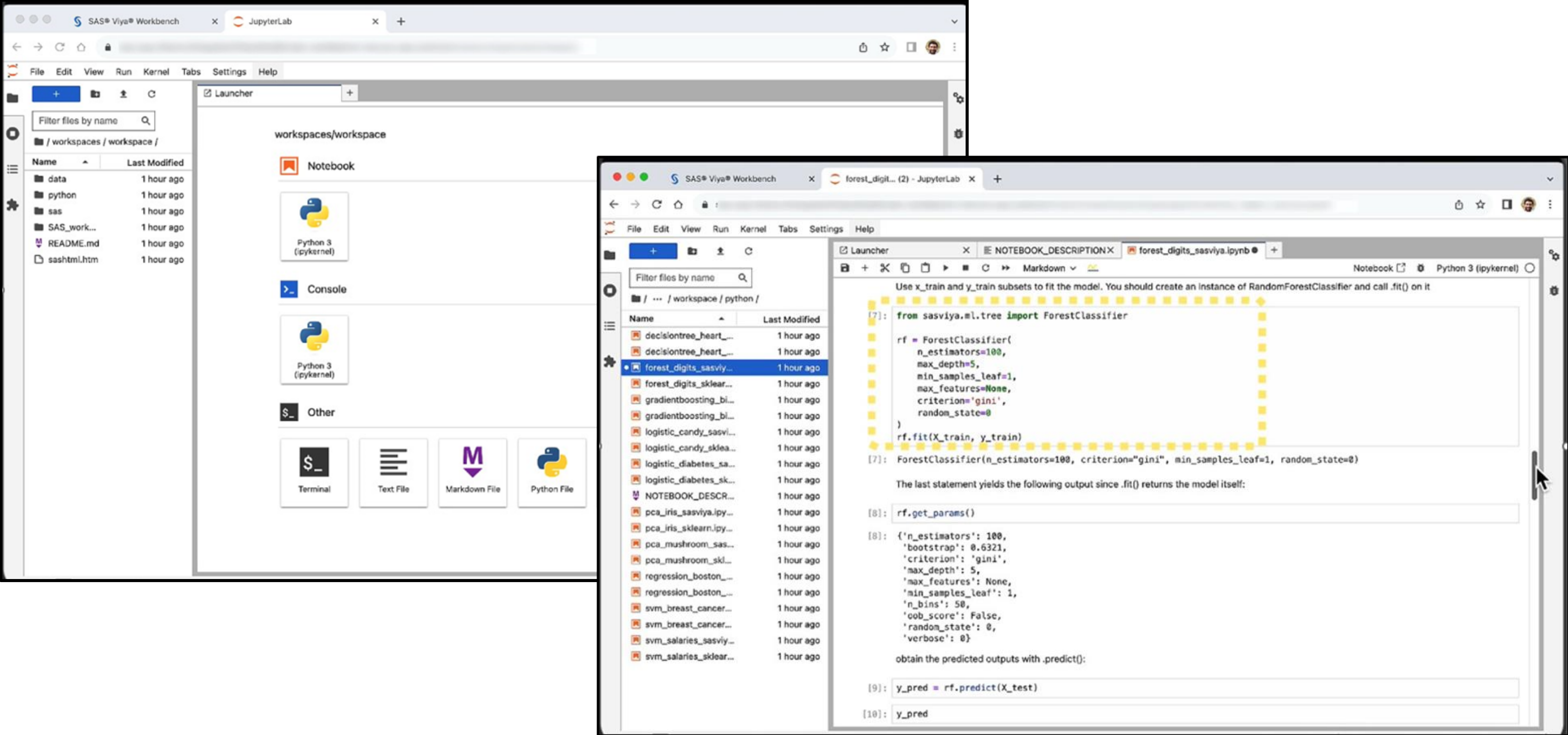
Obs	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered
1	Spring	0	non-working days	Clear + Few clouds + Partly cloudy + Partly cloudy	9.84	14.395	81	0	3	13
2	Spring	0	non-working days	Clear + Few clouds + Partly cloudy + Partly cloudy	9.02	13.635	80	0	8	32
3	Spring	0	non-working days	Clear + Few clouds + Partly cloudy + Partly cloudy	9.02	13.635	80	0	5	27
4	Spring	0	non-working days	Clear + Few clouds + Partly cloudy + Partly cloudy	9.84	14.395	75	0	3	10

- Code from scratch
- Use existing projects or code from SAS 9 or SAS Viya
- Familiar interface



# Python API Integration

SAS Viya Workbench uses the latest Python APIs, enabling retrieval, parsing, updating, and manipulating data



# Provisioning and Storage

Fast deployment and scalable computing options

The screenshot shows a 'New Workbench' dialog box with the following fields and values:

- Name: My Workbench with Azure Storage
- Maximum compute: pup2
- Type: Azure
- Subscription: sas-jomadd
- Storage account: sasjomaddaml1967401128
- Account key: key1
- Share: azureml-filestore-df89b464-0f09-4037-8f6e-0898f0fe1b8f
- Folder: My Azure Storage

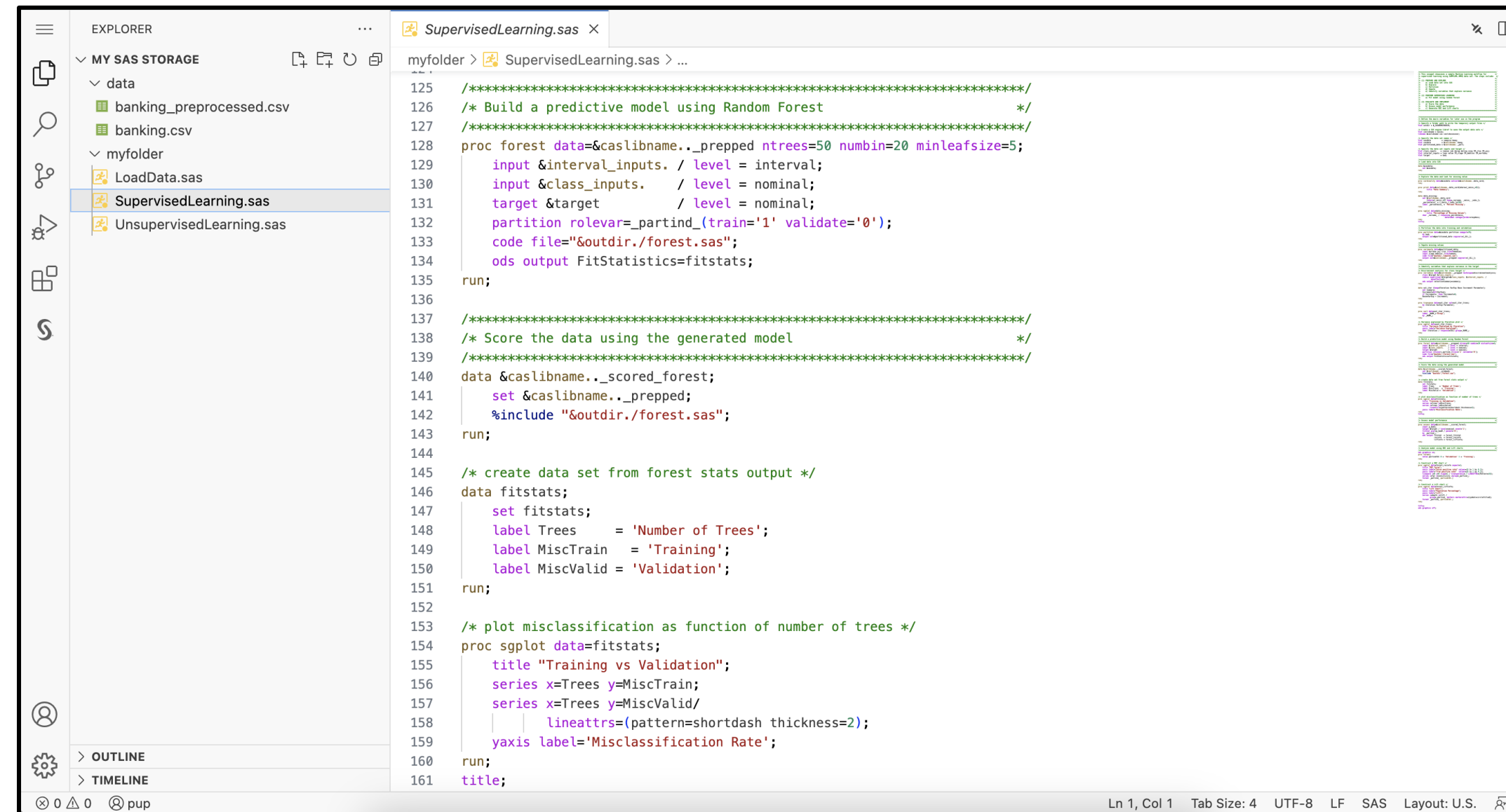
Buttons: OK, Cancel

- Organized file structure to support archiving/backups, updating, collaboration, and compliance
- Choice of compute level for as needed efficiency
- Point-and-click provisioning, requiring minimal IT support.
- Complete structure to support productionizing code.



# Update, Backup, and Team Collaboration

Enhanced developer experience lowers time to production and improves collaboration



The screenshot displays the SAS Studio interface. On the left, the 'EXPLORER' pane shows a file tree under 'MY SAS STORAGE' with folders 'data' and 'myfolder'. The 'SupervisedLearning.sas' file is selected. The main editor shows the SAS code for 'SupervisedLearning.sas', which includes a PROC FOREST step for building a Random Forest model and a PROC SGPLOT step for visualizing misclassification rates. The code is as follows:

```
125 /******  
126 /* Build a predictive model using Random Forest */  
127 /******  
128 proc forest data=&caslibname._prepped ntrees=50 numbin=20 minleafsize=5;  
129     input &interval_inputs. / level = interval;  
130     input &class_inputs. / level = nominal;  
131     target &target / level = nominal;  
132     partition rolevar=_partind(train='1' validate='0');  
133     code file="&outdir./forest.sas";  
134     ods output FitStatistics=fitstats;  
135 run;  
136  
137 /******  
138 /* Score the data using the generated model */  
139 /******  
140 data &caslibname._scored_forest;  
141     set &caslibname._prepped;  
142     %include "&outdir./forest.sas";  
143 run;  
144  
145 /* create data set from forest stats output */  
146 data fitstats;  
147     set fitstats;  
148     label Trees = 'Number of Trees';  
149     label MiscTrain = 'Training';  
150     label MiscValid = 'Validation';  
151 run;  
152  
153 /* plot misclassification as function of number of trees */  
154 proc sgplot data=fitstats;  
155     title "Training vs Validation";  
156     series x=Trees y=MiscTrain;  
157     series x=Trees y=MiscValid/  
158         | lineattrs=(pattern=shortdash thickness=2);  
159     yaxis label='Misclassification Rate';  
160 run;  
161 title;
```

- Utilize SAS code from SAS 9 or SAS Viya
- Run SAS, Python or R
- Update and backup

**Thank you**

