



Graph AI Summit
Amazon SageMaker + TigerGraph

Machine learning for every data scientist and developer

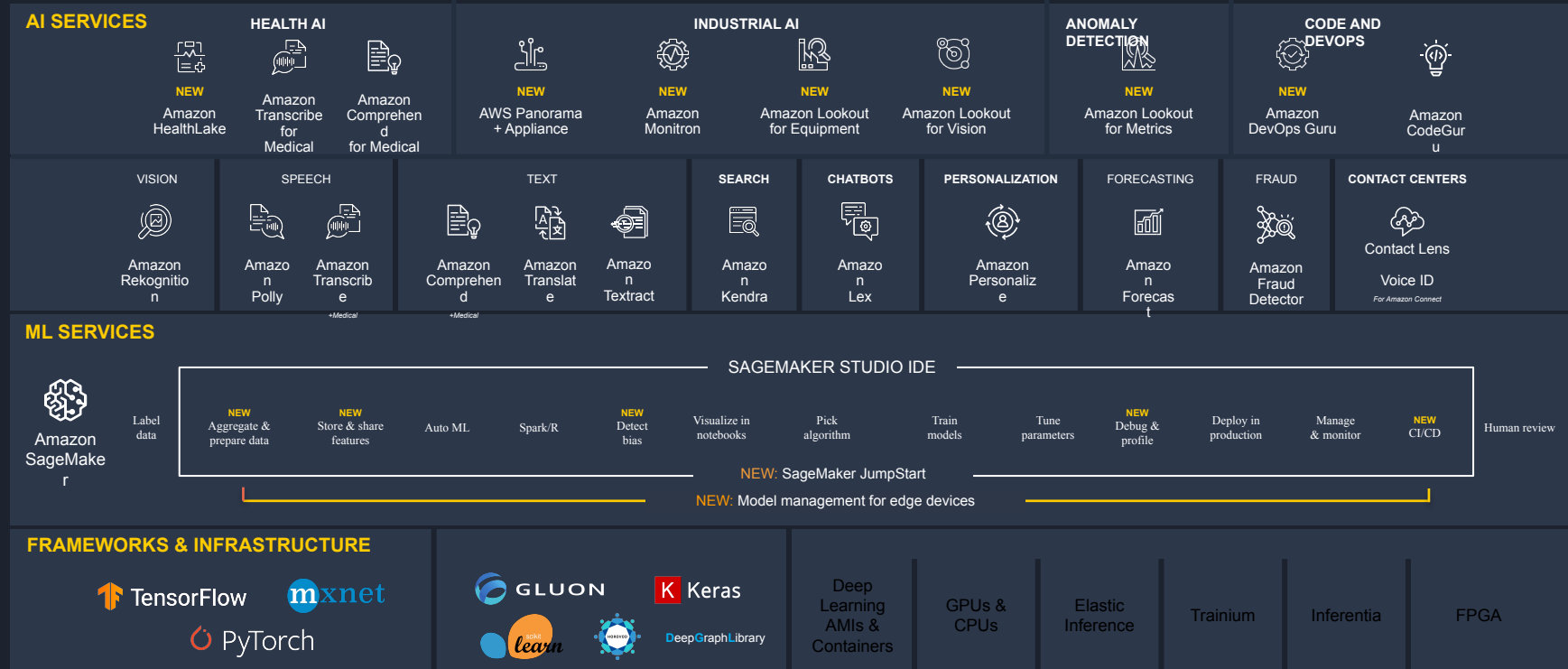
Phi Nguyen
Sr AI/ML Solution Architect

Jon Herke
Manager Developer Relations

Agenda

- SageMaker ML Platform Overview
- SageMaker Studio Notebook Overview
- SageMaker with TigerGraph Integration
- Graph Feature Generation
- Machine Learning Modeling and Deployment

Broadest and most complete set of machine learning capabilities





INCREASED SPENDING

By 2023, spending on AI systems will reach \$97.9B, up 2.5x from \$37.5B in 2019

—IDC



FROM PILOTING TO OPERATIONALIZING

By the end of 2024, 75% of enterprises will shift from piloting to operationalizing AI

—Gartner



ML HANDLES REAL-WORLD TASKS

Driven by advancements in GPUs and compute, availability of data, new algorithms and the cloud

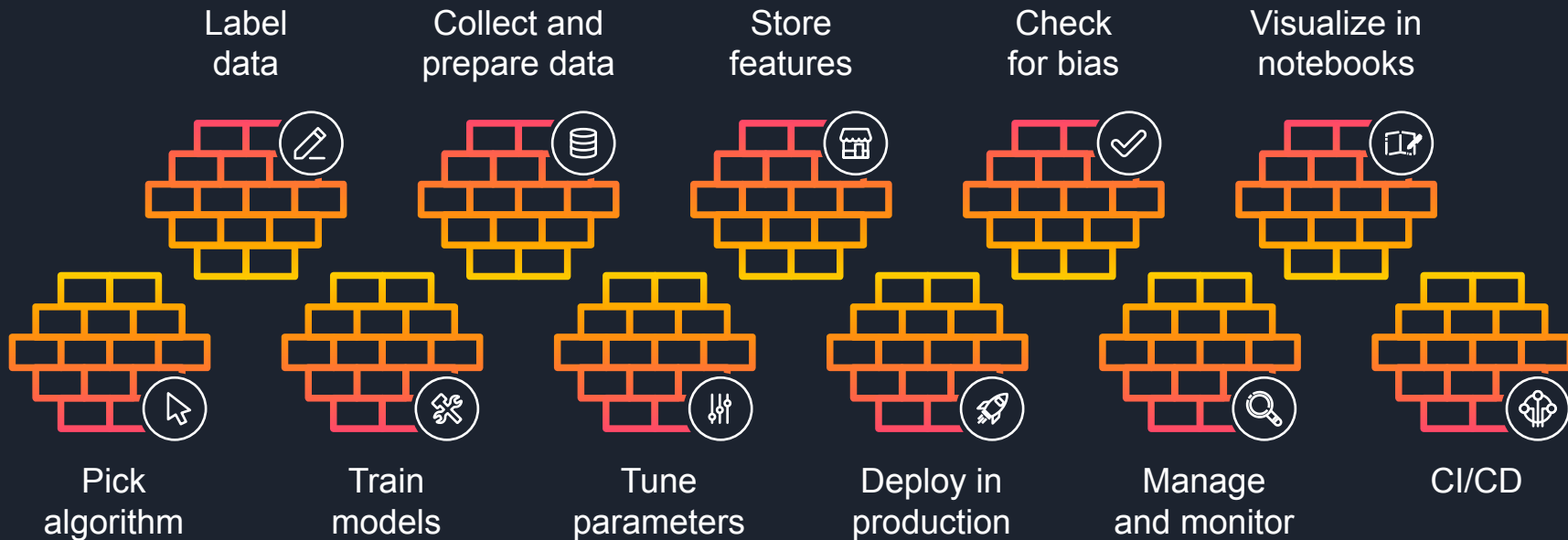


INTEGRATING ML INTO DEVOPS

ML is now part of mainstream DevOps process, not a set of specialized, isolated projects

—Gartner

Machine learning development is complex and costly



SageMaker Studio IDE



MODEL MANAGEMENT FOR EDGE DEVICES

Integrated Workbench

Capabilities designed specifically for ML, data preparation, experiment management, and workflows

Managed Infrastructure

Designed for ultra low latency and high throughput, automatic scaling, and distributed training

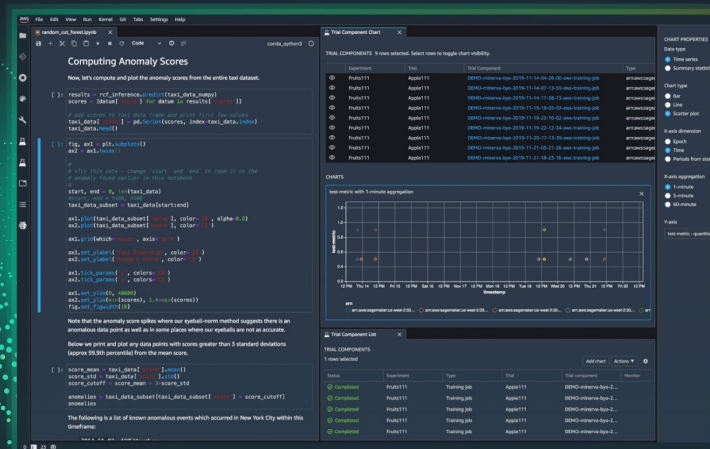
Managed Tooling

Purpose-built from the ground up to work together including auto ML, collaboration, debugger, profiler, bias analyzer, and explainability

<https://aws.amazon.com/sagemaker>

Amazon SageMaker

Most complete, end-to-end ML service



Amazon SageMaker

PREPARE

SageMaker Ground Truth

Label training data for machine learning

SageMaker Data Wrangler **NEW**

Aggregate and prepare data for machine learning

SageMaker Processing

Built-in Python, BYO R/Spark

SageMaker Feature Store **NEW**

Store, update, retrieve, and share features

SageMaker Clarify **NEW**

Detect bias and understand model predictions

BUILD

SageMaker Studio Notebooks

Jupyter notebooks with elastic compute and sharing

Built-in and Bring your-own Algorithms

Dozens of optimized algorithms or bring your own

Local Mode

Test and prototype on your local machine

SageMaker Autopilot

Automatically create machine learning models with full visibility

SageMaker JumpStart **NEW**

Pre-built solutions for common use cases

TRAIN & TUNE

Managed Training

Distributed infrastructure management

SageMaker Experiments

Capture, organize, and compare every step

Automatic Model Tuning

Hyperparameter optimization

Distributed Training Libraries **NEW**

Training for large datasets and models

SageMaker Debugger **NEW**

Debug and profile training runs

Managed Spot Training

Reduce training cost by 90%

DEPLOY & MANAGE

Managed Deployment

Fully managed, ultra low latency, high throughput

Kubernetes & Kubeflow Integration

Simplify Kubernetes-based machine learning

Multi-Model Endpoints

Reduce cost by hosting multiple models per instance

SageMaker Model Monitor

Maintain accuracy of deployed models

SageMaker Edge Manager **NEW**

Manage and monitor models on edge devices

SageMaker Pipelines **NEW**

Workflow orchestration and automation

SageMaker Studio

Integrated development environment (IDE) for ML

Amazon SageMaker key benefits

Most complete,
end-to-end ML service



Accelerate ML development

20+ tools covering the entire ML development lifecycle



Boost data scientist productivity

The world's first integrated development environment (IDE)



Reduce cost

Eliminate costs of writing custom integration code with integrated functionality optimized for ML

ML Builders



Focus on unique business value
Self-service access
Experiment fast
Respond quickly to change



Cloud IT and DevOps



Security
Compliance
Operations
Spend management

Amazon SageMaker

is devops ready



Security

Security features to help you meet strict security requirements of ML workloads



Compliance

PCI, HIPAA, SOC 1/2/3, FedRAMP, and ISO 9001/27001/27017/27018



ML workflows

Create automated workflows in minutes to support thousands of models



Scalability

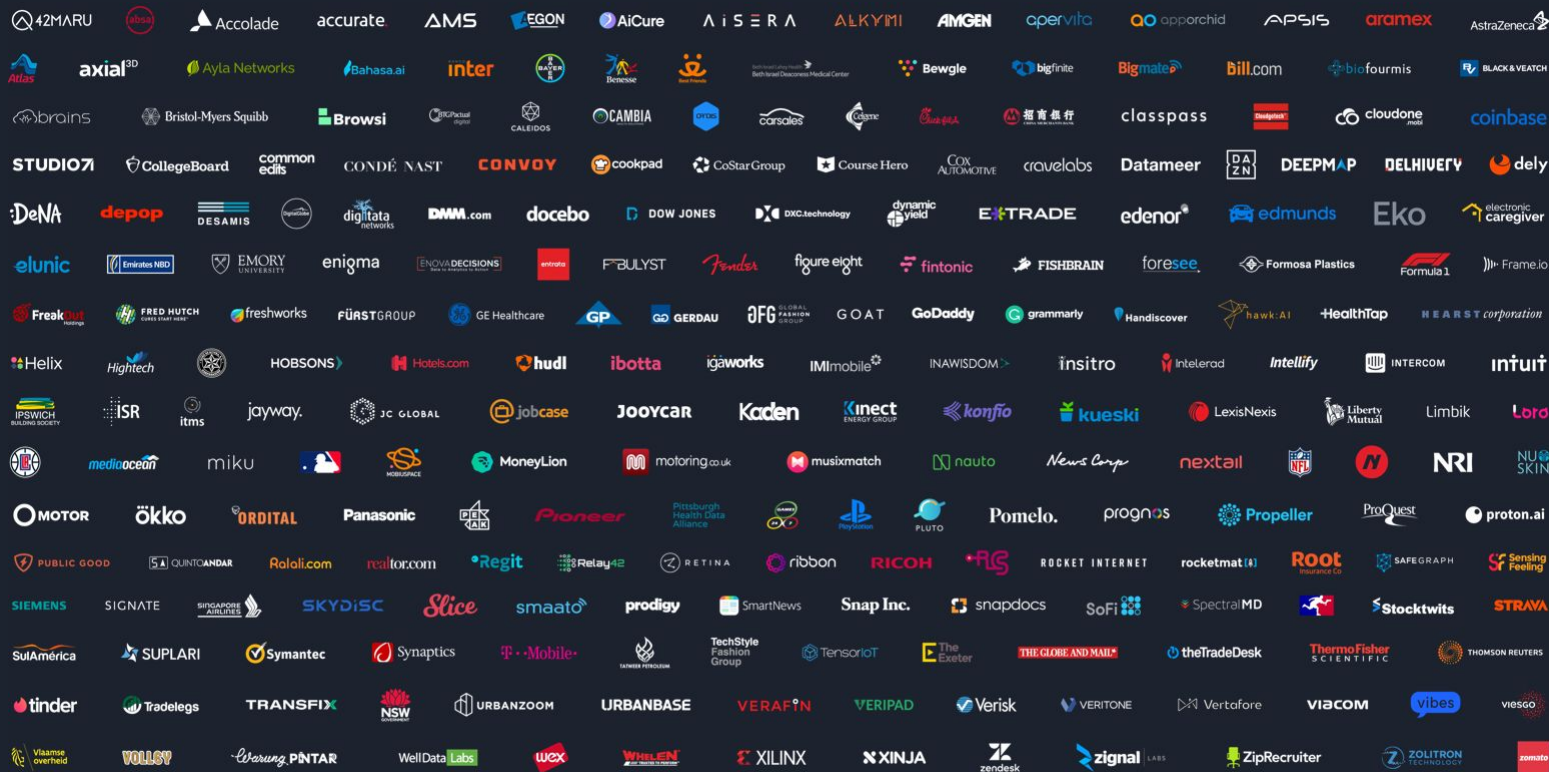
Train complex models with massive datasets



Orchestration

Automatic scheduling and execution of jobs with managed infrastructure

Tens of thousands of customers use **Amazon SageMaker**



SageMaker
Common
use cases



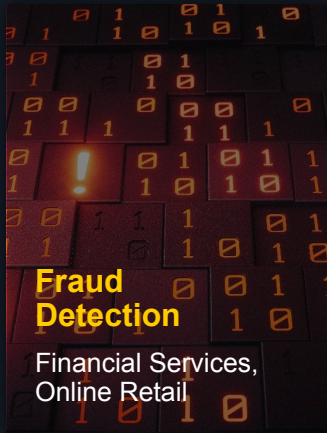
Predictive Maintenance

Manufacturing, Automotive, IoT



Demand Forecasting

Retail, Consumer Goods, Manufacturing



Fraud Detection

Financial Services, Online Retail



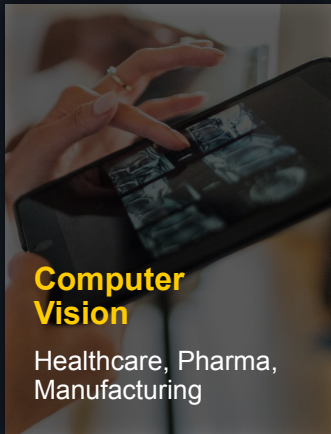
Credit Risk Prediction

Financial Services, Retail



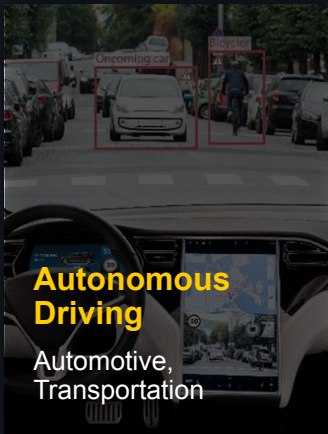
Extract and Analyze Data from Documents

Healthcare, Legal, Media/Ent, Education



Computer Vision

Healthcare, Pharma, Manufacturing



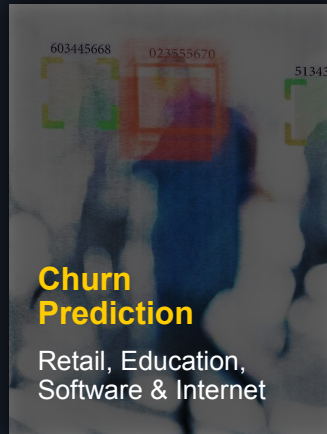
Autonomous Driving

Automotive, Transportation



Personalized Recommendations

Media & Entertainment, Retail, Education



Churn Prediction

Retail, Education, Software & Internet

FAST-START SHARABLE NOTEBOOKS



Easy access with AWS Single Sign-On (SSO)



Fully managed and secure



Fast setup



Easy collaboration



Flexible

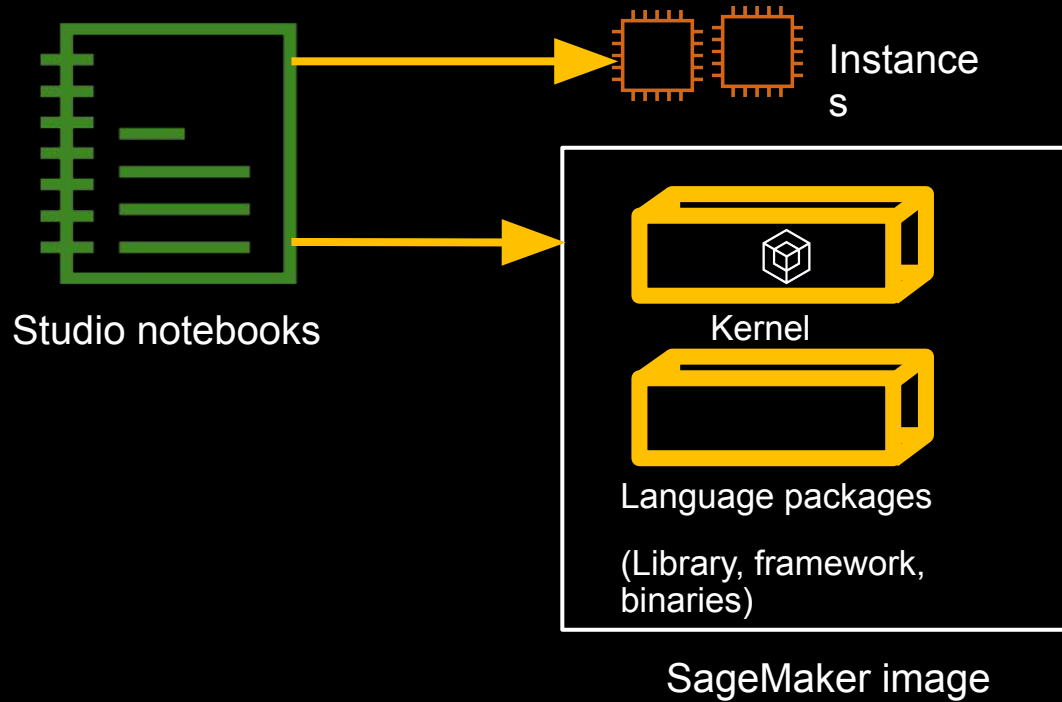
Access your notebooks in seconds

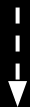
Administrators manage access and permissions

Start your notebooks without configuring compute resources

Share notebooks with a single click

Dial up or down compute resources





- Can enable AWS Identity and Access Management (IAM) or AWS SSO
- Fast startup
- Consolidated experience
- A slice of the shared Amazon Elastic File System (Amazon EFS)



- Low-lift / automatic feature access
 - Autopilot
 - Experiments
 - Debugger
 - Model monitoring
 - Versioning
 - Endpoint management

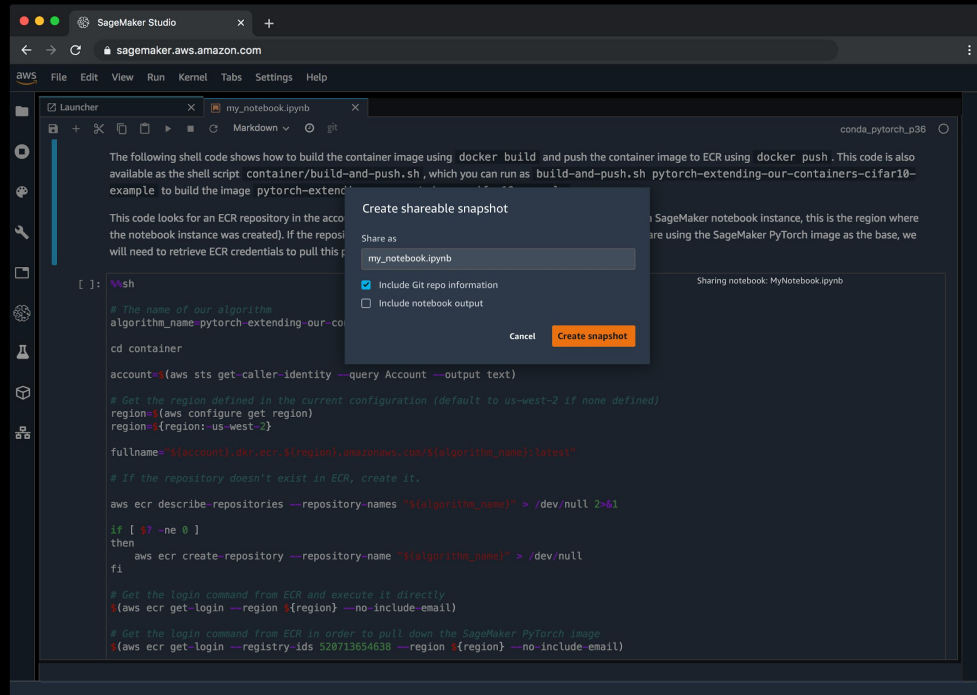


- Can create a shareable link with package dependences in a single click
- Dedicated compute resource
- Elastic compute
- Billing is per usage, not uptime
- Change environments per project

Remember, features are backwards compatible but differ in ease of use across instances vs. Studio

- Extension of the JupyterLab interface
- Dedicated home directory for users
- Quick startup and 1-click deployment
- Dedicated link for users with no direct interface with AWS Management Console
- Get notebook differences and integrate with Git
- Manage resources

Use SageMaker notebooks to easily share work with colleagues



The screenshot displays the SageMaker Studio web interface. The browser address bar shows `sagemaker.aws.amazon.com`. The interface includes a menu bar (AWS, File, Edit, View, Run, Kernel, Tabs, Settings, Help) and a sidebar with navigation icons. The main workspace is divided into a Launcher, a notebook editor, and a terminal. The notebook editor shows a file named `my_notebook.ipynb` with a code cell containing shell commands for building and pushing a Docker container image to ECR. A modal dialog titled "Create shareable snapshot" is open, allowing the user to share the notebook. The dialog includes a "Share as" field with the value `my_notebook.ipynb`, and two checkboxes: "Include Git repo information" (checked) and "Include notebook output" (unchecked). The "Create snapshot" button is highlighted in orange.

```
The following shell code shows how to build the container image using docker build and push the container image to ECR using docker push. This code is also available as the shell script container/build-and-push.sh, which you can run as build-and-push.sh pytorch-extending-our-containers-cifar10-example to build the image pytorch-extending-our-containers-cifar10-example.
```

```
This code looks for an ECR repository in the account where the notebook instance was created. If the repository does not exist, it will create it. If the repository exists, you will need to retrieve ECR credentials to pull this image.
```

```
[ ]: $!sh
```

```
# The name of our algorithm
algorithm_name=pytorch-extending-our-co

cd container

account=$(aws sts get-caller-identity --query Account --output text)

# Get the region defined in the current configuration (default to us-west-2 if none defined)
region=$(aws configure get region)
region=${region:-us-west-2}

fullname=${account}.dkr.ecr.${region}.amazonaws.com/${algorithm_name}:latest

# If the repository doesn't exist in ECR, create it.
aws ecr describe-repositories --repository-names "${algorithm_name}" > /dev/null 2>&1

if [ $? -ne 0 ]
then
  aws ecr create-repository --repository-name "${algorithm_name}" > /dev/null
fi

# Get the login command from ECR and execute it directly
$(aws ecr get-login --region ${region} --no-include-email)

# Get the login command from ECR in order to pull down the SageMaker PyTorch image
$(aws ecr get-login --registry-ids 520713654630 --region ${region} --no-include-email)
```

Sharing notebook: MyNotebook.ipynb

Create shareable snapshot

Share as
my_notebook.ipynb

Include Git repo information
 Include notebook output

Cancel Create snapshot

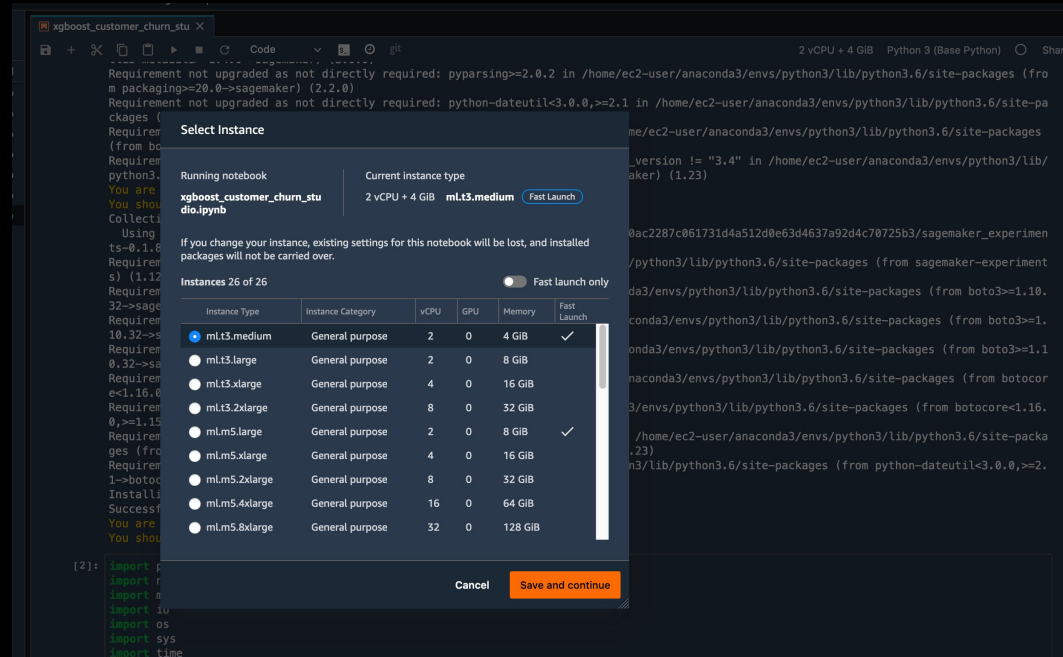
Base execution environment and notebook output are captured

The screenshot shows a Jupyter Notebook window titled 'xgboost_customer_churn_stu'. The top status bar indicates 'Kernel: CPU: 0.00% MEM: 2.84% 2 vCPU + 4 GiB Python 3 (Base Python)'. The notebook content includes a data table with 20 columns and 2333 rows. A modal dialog box is open in the center, titled 'Create shareable snapshot', with a green success message: 'Successfully created snapshot of xgboost_customer_churn_studio.ipynb'. Below the message, it provides a 'Shareable link' and a 'Copy link' button. The link text is partially visible as 'https://... .us-west-2.s3'. A 'Close' button is at the bottom of the dialog. Below the dialog, the notebook code is visible, showing the upload logic:

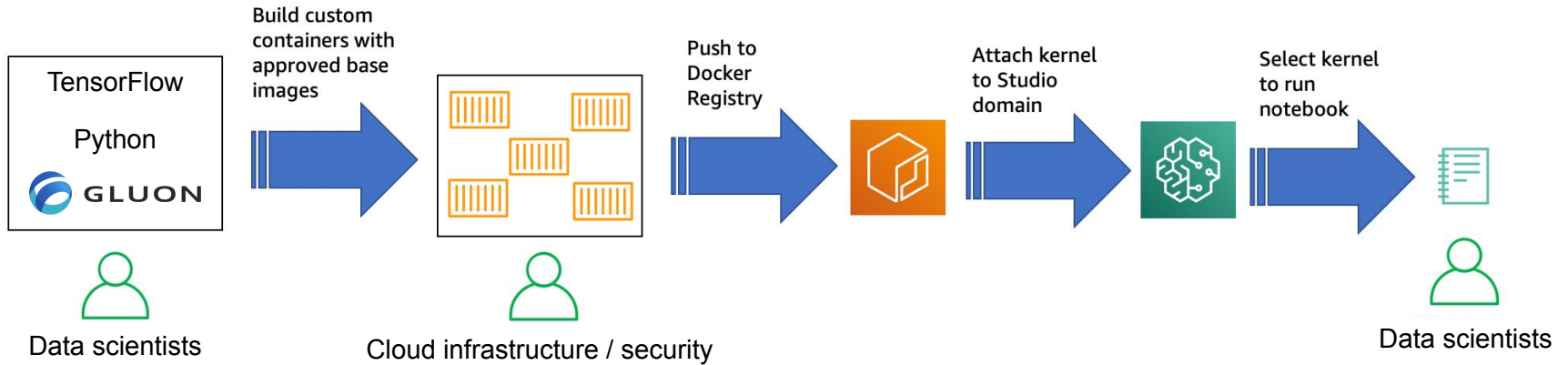
```
[4]:  
Churn  Account Length  VMail Message  Day Mins  Day Calls  Eve Mins  Eve Calls  Night Mins  Night Calls  Intl Mins  Intl Calls  CustServ Calls  State_AK  State_AL  State_AR  State_AZ  State_CA  State_CO  State_CT  State_DC  Stat  
0 0 106 0 274.4 120 198.6 82 160.8 62 6.0 3 1 0 0 0 0 0 0 0 0  
1 0 28 0 187.8 94 248.6 86 208.8 124 10.6 5 0 0 0 0 0 0 0 0 0  
2 1 148 0 279.3 104 201.6 87 280.8 99 7.9 2 2 0 0 0 0 0 0 0 0  
3 0 132 0 191.9 107 206.9 127 272.0 88 12.6 2 1 0 0 0 0 0 0 0 0  
4 0 92 29 155.4 110 188.5 104 254.9 118 8.0 4 3 0 0 0 0 0 0 0 0  
... ..  
2328 0 106 0 194.8 133 213.4 73 190.8 92 11.5 7 0 0 0 0 0 0 0 0 0  
2329 1 125 0 201.6 104 201.6 87 280.8 99 7.9 2 4 0 0 0 0 0 0 0 0  
2330 0 129 0 191.9 107 206.9 127 272.0 88 12.6 2 4 0 0 0 0 0 0 0 0  
2331 0 159 0 191.9 107 206.9 127 272.0 88 12.6 2 5 0 0 0 0 0 0 0 0  
2332 0 99 0 191.9 107 206.9 127 272.0 88 12.6 2 2 0 0 0 0 0 0 0 0  
2333 rows x 70 columns  
  
Now we'll upload the data to S3 if one does not already exist.  
identity()['Account']  
  
[8]:  
account_id = sess.get_account_id()  
bucket = 'sagemaker-us-west-2-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx'  
prefix = 'xgboost-churn'  
  
try:  
    if sess.region_name == "us-west-2":  
        sess.client('s3').create_bucket(Bucket=bucket)  
    else:  
        sess.client('s3').create_bucket(Bucket=bucket,
```

SageMaker Studio notebooks provide flexible compute

- Switch to compute with more memory, more CPU, or GPU acceleration when you need it
- Switch back to very inexpensive compute when you want



SageMaker Studio notebooks
provide persistent custom development environment



SageMaker Studio does security hardening

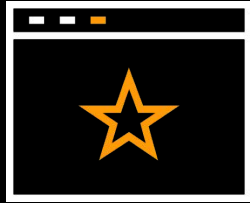


VPC support

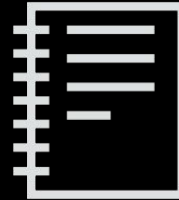


End-to-end encryption

SageMaker Studio notebooks provide full control of the development environment



Bring your own image



View and shut down
instances in Studio

Demo

