



# Compute

Introducing

# AWS Graviton2 Processor

Enabling the best price/performance for your cloud workloads

## Graviton1 Processor



First ARM-based processor  
in major cloud

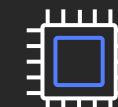


Built on 64-bit ARM Neoverse cores  
with AWS-designed 16 nm silicon



Up to 16 vCPUs, 10 Gbps enhanced  
networking, 3.5 Gbps EBS bandwidth

## Graviton2 Processor



Built with 64-bit ARM Neoverse  
cores with AWS-designed  
7 nm silicon process



Up to 64 vCPUs, 20 Gbps enhanced  
networking, 14 Gbps EBS bandwidth



7x performance, 4x compute cores,  
and 5x faster memory

Introducing

# AWS Graviton2 Based Instances

Up to 40% better price-performance for general purpose, compute intensive, and memory intensive workloads.

## M6g

Built for: General-purpose workloads such as application servers, mid-size data stores, and microservices

Instance storage option: M6gd

## C6g

Built for: Compute intensive applications such as HPC, video encoding, gaming, and simulation workloads

Instance storage option: C6gd

## R6g

Built for: Memory intensive workloads such as open-source databases, or in-memory caches

Instance storage option: R6gd

---

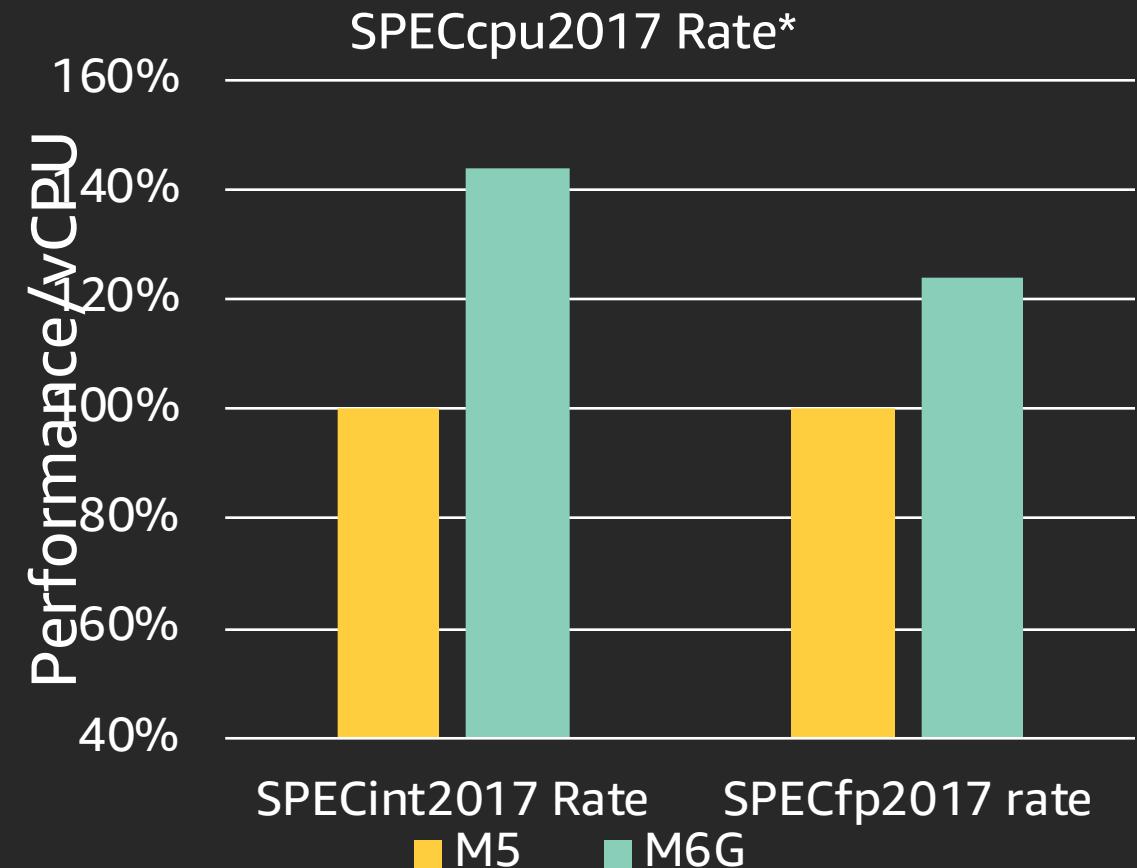
M6g Available  
in Preview

Coming in 2020



# SPEC cpu2017

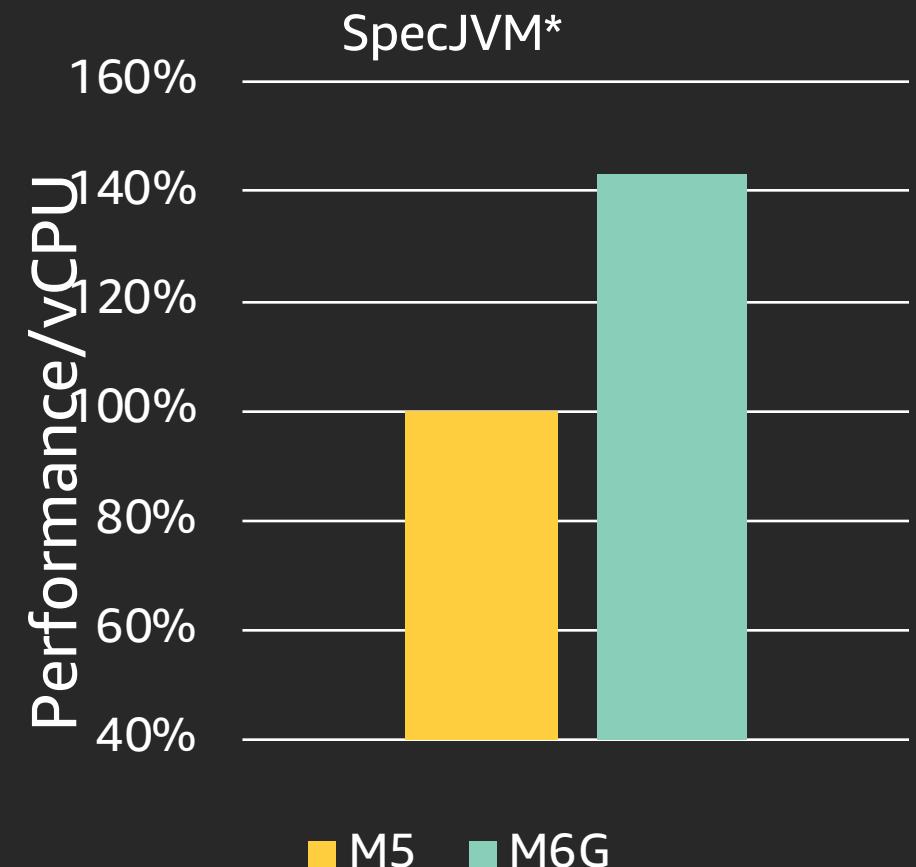
- Industry standard CPU intensive benchmark
- Run on all vCPUs concurrently
- Comparing performance/vCPU



\* All SPEC scores estimates, compiled with GCC9 -O3 -march=native, run on largest single socket size for each instance type tested.

# SPEC jvm2008

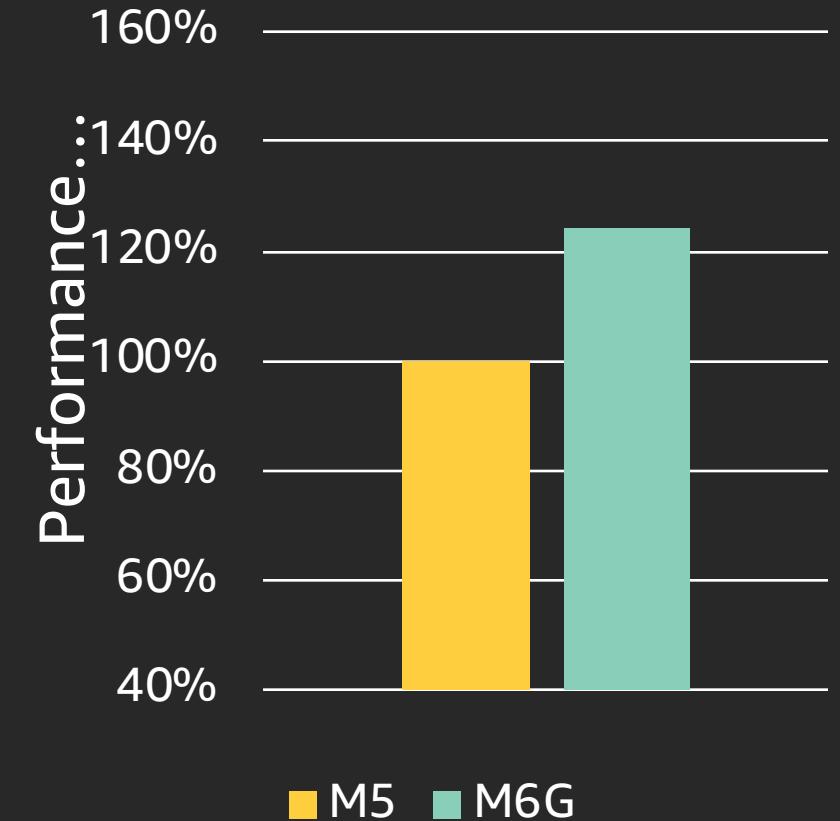
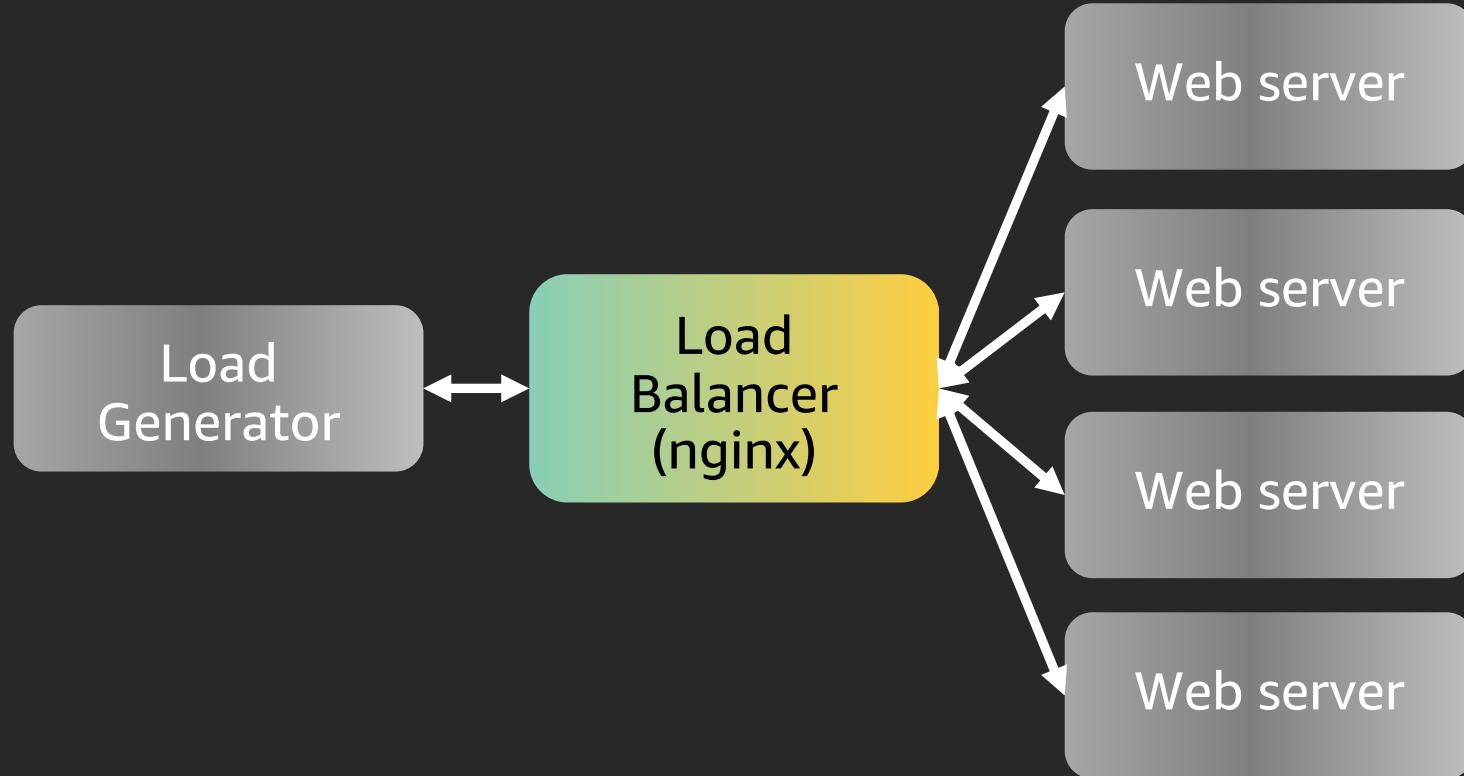
- Java VM benchmark
- Run across all vCPUs concurrently
- Comparing performance/vCPU



\* All SPEC scores estimates, run with OpenJDK11 and skipping compiler\* and startup.\* tests  
Tests run on largest single-socket instance size for each instance type tested.



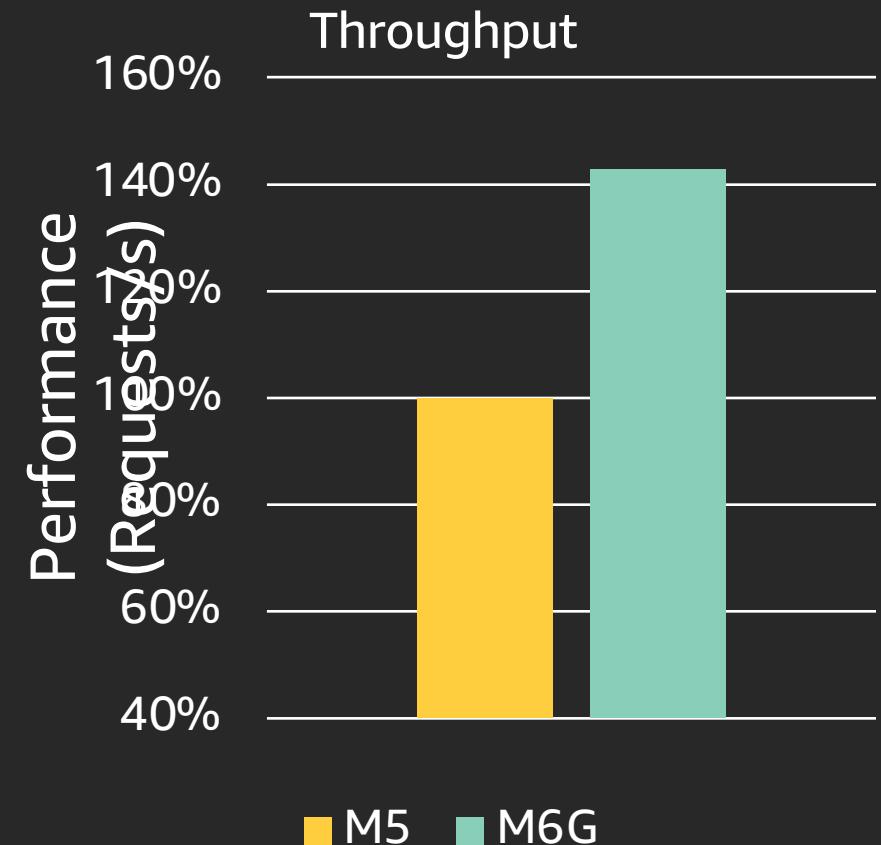
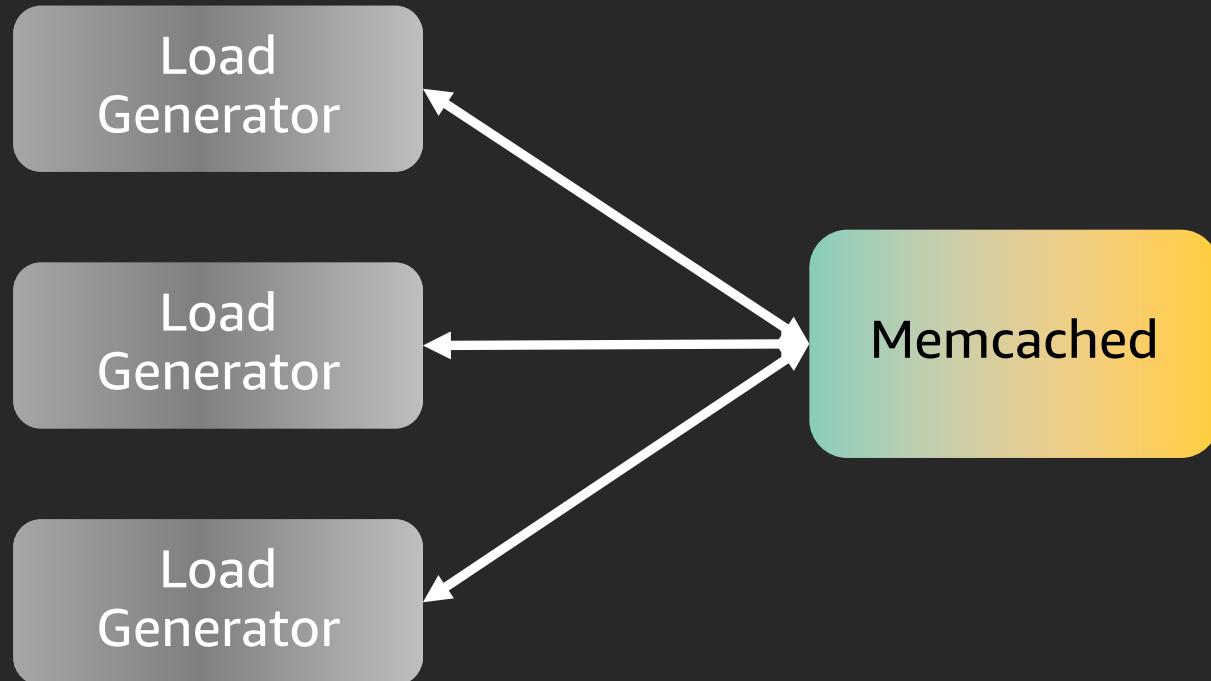
# Load Balancing with Nginx



NGINX v1.15.9, 512 clients, 128 GET/POST payloads, all HTTPS connections, AES128-GCM-SHA256, OpenSSL 1.1.1, 4 target machines, all tests run on 4xl size; load generator c5.9xl; web servers c5.4xl; All servers run in a cluster placement group



# Memcached

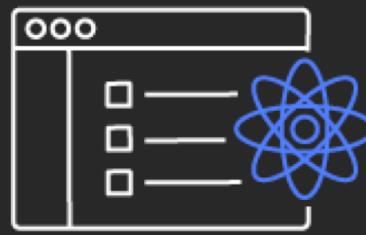


Memcached v1.5.16, 16B keys, 128B values, 7.8M KV-pairs, 576 connections for load generation from 2x c5.9xlarge instances, 16 additional connections used to measure latency from 1 additional c5.9xlarge; each connection maintains 4096 outstanding requests; All servers in a cluster placement group

Introducing

# Amazon Braket

Fully managed service that makes it easy for scientists and developers to explore and experiment with quantum computing.



Single environment  
to design, test, and  
run quantum  
algorithms



Experiment with a  
variety of quantum  
hardware  
technologies



Run hybrid  
quantum and  
classical algorithms



Get Expert Help

Introducing

# AWS Compute Optimizer

Identify optimal Amazon EC2 instances and EC2 Auto Scaling group for your workloads using a ML-powered recommendation engine



Easily identify optimal AWS resources for your workloads



Supports over 140 EC2 instance type options in the M, C, R, T, and X instance families



Lower costs and increase performance for your AWS workloads

[Dashboard](#)[▼ Recommendations per service](#)[EC Instances](#)[AutoScaling Groups](#)

## Recommendation detail: Portal1\_DEV [Info](#)

[View in EC2 console](#)[Export](#)

### Configuration comparison and selection [Info](#)

Select one of the recommended options to compare it against your current utilization metrics below.

Configuration	Instance type	On-Demand price	Price difference	Risk level	vCPUs	Memory	Storage	Net
Current (Portal2_P)	m5.2xlarge	\$0.384 per hour			8	32GiB	EBS Only	Up
<input checked="" type="radio"/> Option 1	m5.xlarge	\$0.192 per hour	-\$0.192 per hour	Low	4	16GiB	EBS Only	Up
<input type="radio"/> Option 2	t3.xlarge	\$0.1664 per hour	-\$0.2176 per hour	Low	4	16GiB	EBS Only	Mo
<input type="radio"/> Option 3	r5.large	\$0.126 per hour	-\$0.258 per hour	Medium	2	16GiB	FRS Only	Un

### Option 1 (m5.xlarge) v. Current (m5.2xlarge) [Info](#)

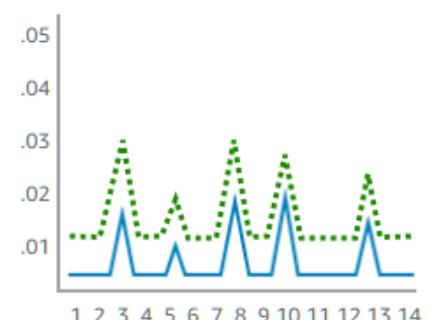
[Statistic: Average](#)[Time range: Last 2 weeks](#)

Key:

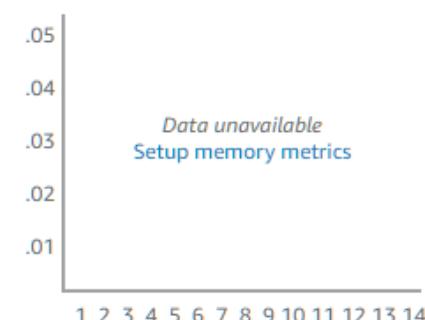
— Current utilization

... Recommended projected utilization

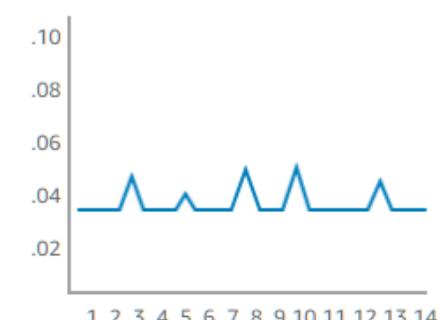
#### CPU utilization (percent)



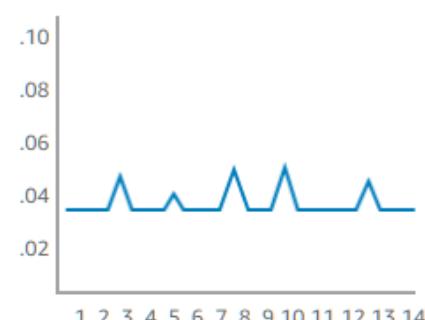
#### Memory utilization (percent)



#### Network in (Bytes)



#### Network Out (Bytes)



IS

ential

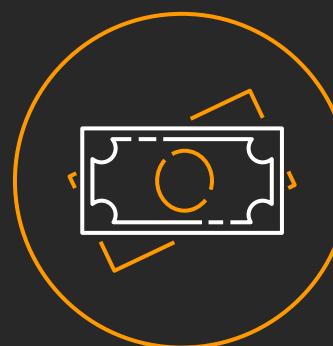
Announced – November 6

# Savings Plans

Simplify purchasing with a flexible pricing model that offers savings of up to 72% on Amazon ECS, AWS Fargate & AWS Lambda usage



Receive lower rates automatically. **Easy to use** with recommendations in AWS Cost Explorer



Significant savings of up to 72%



Flexible across instance family, size, OS, tenancy or AWS Region; also applies to AWS Fargate & soon to AWS Lambda usage



Introducing

General Availability – December 3

# Amazon EKS for AWS Fargate

The only way to run serverless Kubernetes containers securely, reliably, and at scale



---

Simplified deployment,  
management, and scaling  
of Kubernetes on AWS



---

Strong security isolation  
for every pod by default



---

Focus on building  
applications



# Storage

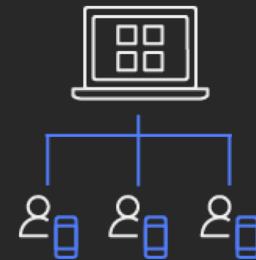
Introducing

# Amazon S3 Access Points

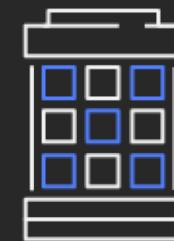
Simplify managing data access at scale for applications using shared data sets on Amazon S3. Easily create hundreds of access points per bucket, each with a unique name and permissions customized for each application.



Simple to create custom access  
to shared data sets on same bucket



Add access points  
as application set grows



Use access points to easily  
restrict access to VPC

# Databases & Analytics

Introducing

Preview – December 3

# Amazon Managed Apache Cassandra Service

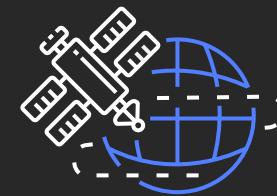
A scalable, highly available, and serverless Apache Cassandra-compatible database service. Run your Cassandra workloads in the AWS cloud using the same Cassandra application code and developer tools that you use today.



Apache Cassandra-compatible



No servers to manage



Performance at scale



Highly available and secure



# Amazon Aurora Machine Learning Integration

Simple, optimized, and secure Aurora, SageMaker, and Comprehend (in preview) integration. Add ML-based predictions to databases and applications using SQL, without custom integrations, moving data around, or ML experience.



ML predictions  
on relational data



Integration with  
SageMaker &  
Comprehend



Familiar SQL  
language, no  
ML expertise



Low-latency,  
real-time

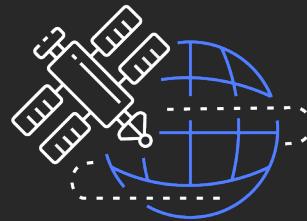


Security &  
governance

Introducing

# Amazon RDS Proxy

Fully managed, highly available database proxy feature for Amazon RDS. Pools and shares connections to make applications more scalable, more resilient to database failures, and more secure.



Scalable applications  
with database efficiency



Resilient applications  
with fast failover



Secure applications  
with data protection

# Security & Networking

Introducing

Announced – November 21

# CloudTrail Insights

Identify unusual activity in your AWS accounts



- Unexpected spikes in resource provisioning
  - Bursts of IAM management actions
  - Gaps in periodic maintenance activity
- ✓ Save time sifting through logs
  - ✓ Get ahead of issues before they impact your business



Introducing

# AWS Detective

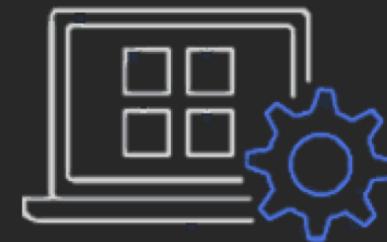
Quickly analyze, investigate, and identify the root cause of security findings and suspicious activities.



Automatically distills  
& organizes data into  
a graph model



Easy to use visualizations  
for faster & effective  
investigation



Continuously updated as  
new telemetry becomes  
available

Introducing

# AWS IAM Access Analyzer

Continuously ensure that policies provide the intended public and cross-account access to resources, such as Amazon S3 buckets, AWS KMS keys, & AWS Identity and Access Management roles.



## Quickly analyze resource policies

Analyzes resource policies for public or cross-account access



## Continuously monitor and analyze permissions

Analyzes new or updated resource policies to help you understand potential security implications



## Provides the highest levels of security assurance

Uses automated reasoning, a form of mathematical logic, to determine all possible access paths allowed by a resource policy

# Improving the Developer Experience

# Provisioned Concurrency on AWS Lambda

- Keeps functions initialized and hyper-ready, ensuring start times stay in the milliseconds
- Builders have full control over when provisioned concurrency is set
- No code changes are required to provision concurrency on functions in production
- Can be combined with AWS Auto Scaling at launch



Introducing

Preview – December 4

# HTTP APIs for Amazon API Gateway

Achieve up to 67% cost reduction and 50% latency reduction compared to REST APIs. HTTP APIs are also easier to configure than REST APIs, allowing customers to focus more time on building applications.



Reduce application costs by  
up to 67%



Reduce application latency by  
up to 50%



Configure HTTP APIs easier  
and faster than before

# Extending AWS beyond the Region

New Feature

Announced – November 25

# Alexa Voice Service (AVS) Integration for IoT Core

Quickly and cost effectively go to market with Alexa built-in capabilities on new categories of products such as light switches, thermostats, and small appliances.



---

Lowers the cost of integrating Alexa Voice up to 50% by reducing the compute and memory footprint required

---

Build new categories of Alexa Built-in products on resource constrained devices (e.g. ARM 'M' class microcontrollers with <1MB embedded RAM).



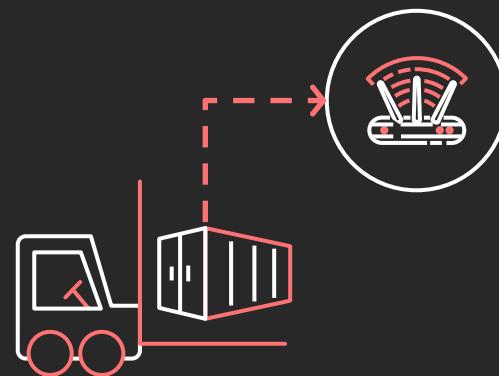
---

Accelerate time to market with certified partner development kits that work with AVS Integration for IoT Core by default.

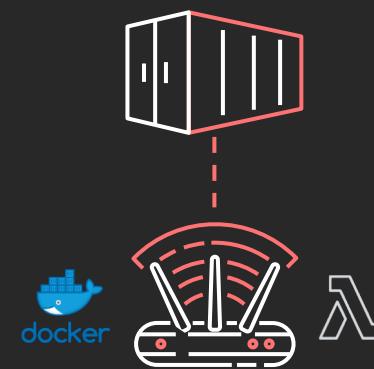


# Container Support for AWS IoT Greengrass

Deploy containers seamlessly to edge devices



Move containers from the cloud to edge devices using AWS IoT Greengrass, without rewriting any code.



Enables both Docker & AWS Lambda components to operate seamlessly together at the edge



Use AWS IoT Greengrass Secrets Manager to manage credentials for private container registries.



Now Available

General Availability – December 3

# AWS Outposts

Fully managed service that extends AWS infrastructure, AWS services, APIs, and tools to virtually any connected customer site. Truly consistent hybrid experience for applications across on-premises and cloud environments. Ideal for low latency or local data processing application needs.



**Same AWS-designed infrastructure**  
as in AWS regional data centers  
(built on AWS Nitro System)  
delivered to customer facilities



**Fully managed, monitored, and  
operated by AWS**  
as in AWS Regions



**Single pane of management**  
in the cloud providing the  
**same APIs and tools** as  
in AWS Regions



# AWS Outposts



# AI Services

# Human In the Loop

# Customers are forced to choose



OR



ML only systems are high speed and low cost, but do not support nuanced decision making

Human only workflows offer nuanced decision making, but they're low speed and high cost.

Customers need



Machine Learning and humans working together

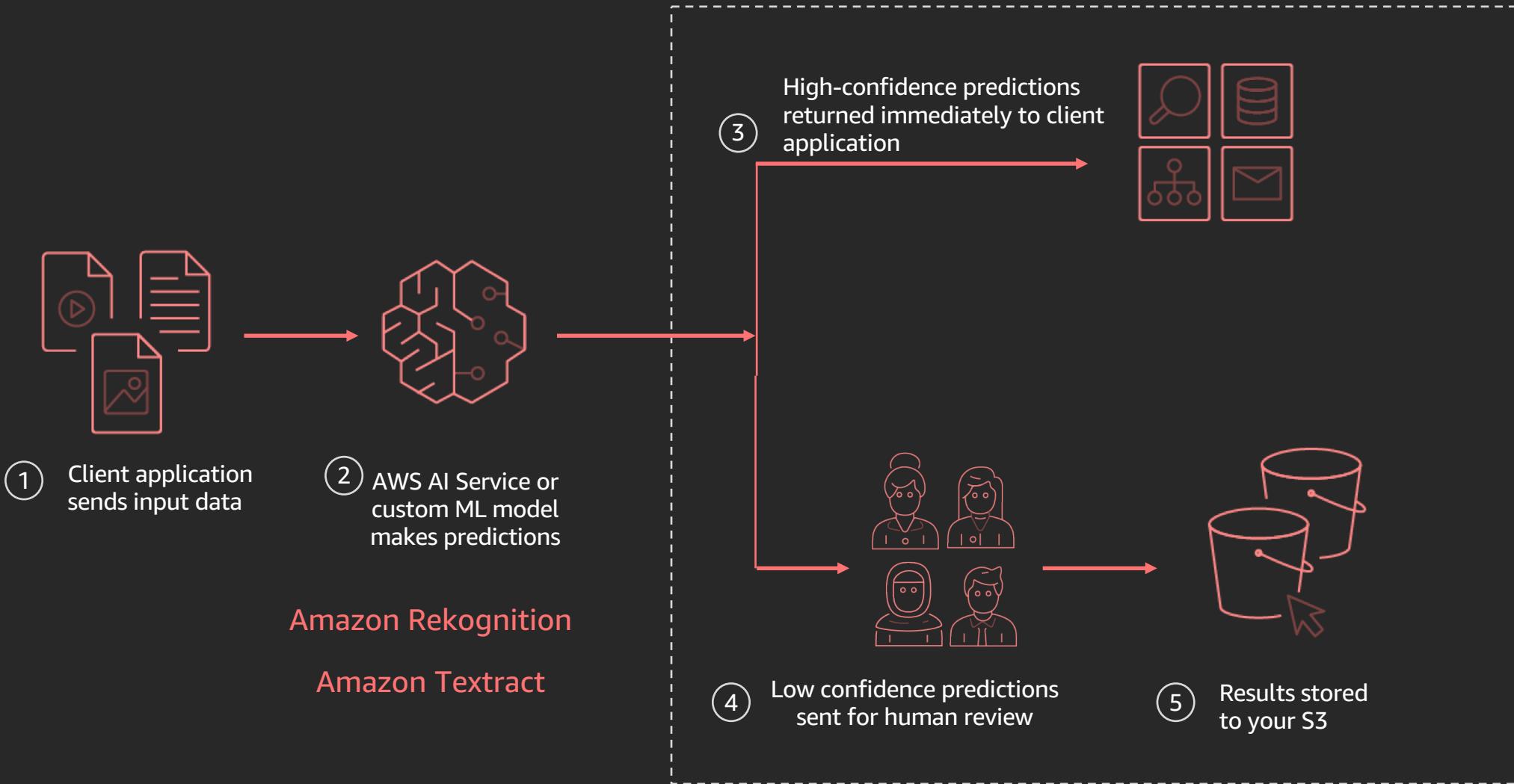
# Introducing Amazon Augmented AI (A2I)

---

A2I lets you easily implement **human review** in machine learning **workflows** to improve the accuracy, speed, and scale of complex decisions.

---

# How Amazon Augmented AI works

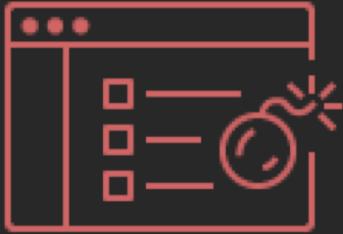


# Fraud Detection

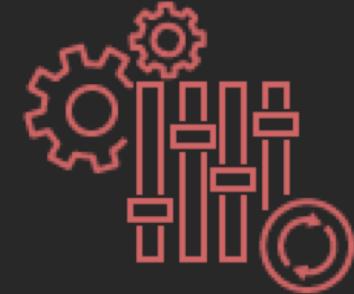
# Fraud detection is difficult



\$\$\$ billions lost to  
fraud each year



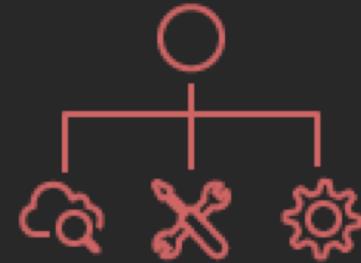
Online business prone  
to fraud attacks



Bad actors often  
change tactics



Changing rules =  
more human reviews



Dependent on others to  
update detection logic

# Fraud detection with ML is also difficult



Top data scientists are  
costly & hard to find



One-size-fits-all models  
underperform



Often need to  
supplement data



Data transformation +  
feature engineering



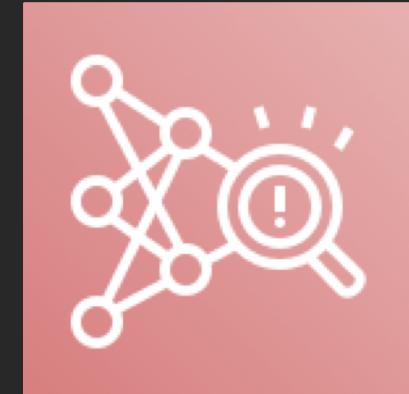
Fraud imbalance =  
needle in a haystack

# Introducing Amazon Fraud Detector

---

A **fraud detection service** that makes it easy for businesses to use machine learning to detect online fraud in **real-time, at scale**

---



# Amazon Fraud Detector – Key Features



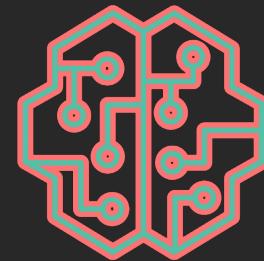
Pre-built fraud  
detection model  
**templates**



Automatic  
creation of  
custom fraud  
detection  
models



Models learn  
from past  
attempts to  
defraud **Amazon**



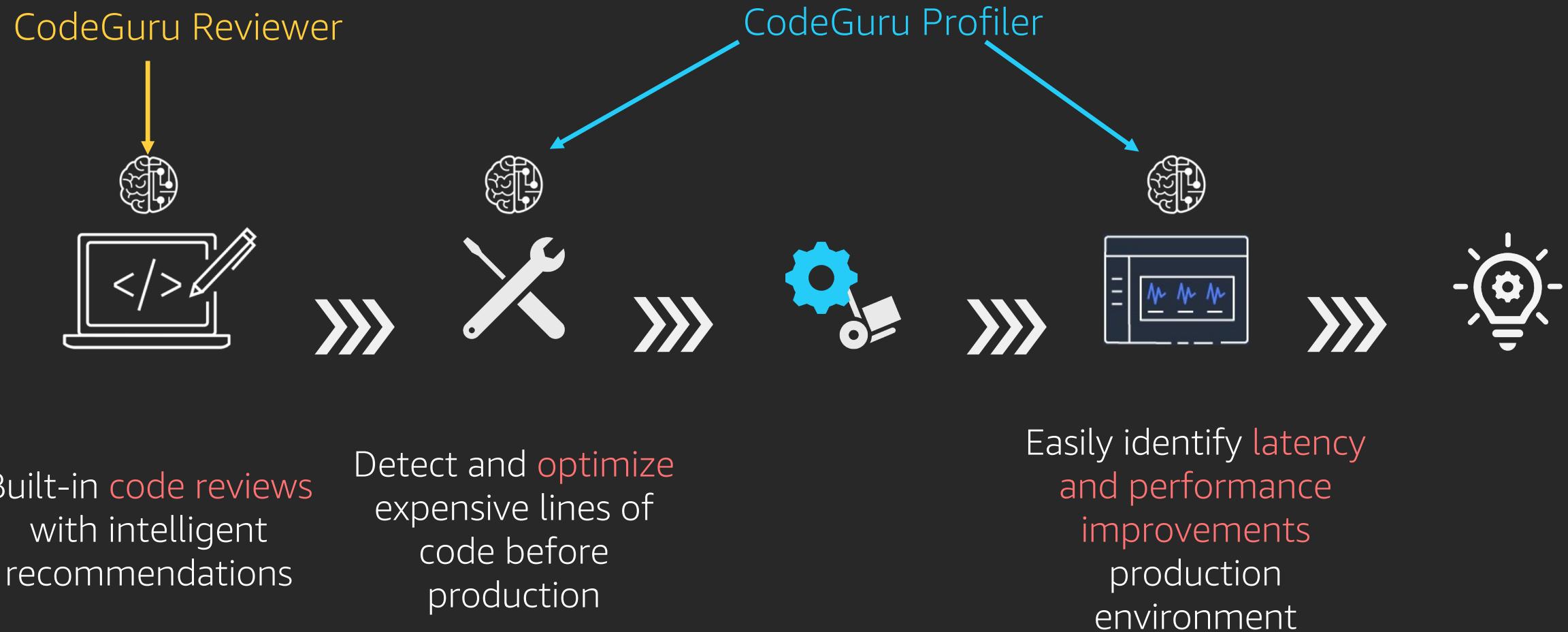
Amazon  
**SageMaker**  
integration



One interface to  
review past  
**evaluations** and  
detection **logic**

# Improving code quality

# Introducing AWS CodeGuru



# CodeGuru Reviewer: How It Works

Customer provides source code as input

Java  
AWS CodeCommit  
Github



Extract semantic features / patterns



ML algorithms identify similar code for comparison



Customers see recommendations as Pull Request feedback



Input:  
Source Code

Feature Extraction

Machine Learning

Output:  
Recommendations

# CodeGuru Profiler: How It Works

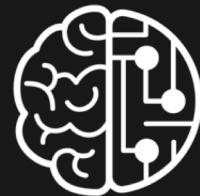
Customer application runs in production



CodeGuru Profiler continuously captures application stack trace information



CodeGuru Profiler detects performance inefficiencies in the live application



Customers see recommendations in their automated efficiency reports and visualizations



Input:  
Live application stack trace

Application profile sampling

Pattern matching

Output:  
Method names,  
Recommendations  
and searchable  
visualizations

# ML services

# Introducing Amazon SageMaker Studio

The first fully integrated development environment (IDE) for machine learning

---



Collaboration at scale



Easy experiment management



Automatic model generation



Higher quality ML models



Increased productivity

Share scalable notebooks without tracking code dependencies

Organize, track, and compare thousands of experiments

Get accurate models for with full visibility & control without writing code

Automatically debug errors, Code, build, train, deploy, & monitor models, & maintain high quality monitor in a unified visual interface

Amazon SageMaker Studio File Edit View Run Kernel Git Tabs Settings Help

xgboost\_customer\_churn.ipynb X

Markdown conda\_amazonei\_mxnet\_p27

- Have the predictor variable in the first column
- Not have a header row

But first, let's convert our categorical features into numeric features.

```
[ ]: model_data = pd.get_dummies(churn)
model_data = pd.concat([model_data['Churn?_True.'], model_data.drop(['Chur
< ... >
```

And now let's split the data into training, validation, and test sets. This will help prevent us from overfitting the model, and allow us to test the models accuracy on data it hasn't already seen.

```
[ ]: train_data, validation_data, test_data = np.split(model_data.sample(frac=1)
train_data.to_csv('train.csv', header=False, index=False)
validation_data.to_csv('validation.csv', header=False, index=False)
< ... >
```

Now we'll upload these files to S3.

```
[ ]: boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix,
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix,
< ... >
```

Trial Component Chart

A line chart titled "Trial Component Chart" showing training loss over time. The y-axis is labeled "trainloss\_last" and ranges from 0.0 to 0.4. The x-axis is labeled "period" and shows values 0, 1, 2, 3, 4, 5, 6, 7, 8. There are four data series: a blue line starting at ~0.35 and ending at ~0.15; a red line starting at ~0.25 and ending at ~0.05; a green line starting at ~0.38 and ending at ~0.15; and a yellow line starting at ~0.15 and ending at ~0.05. All lines show a general downward trend.

Trial Component List

Status	Experiment	Type	Trial	...
Completed	customer-churn-predic...	Training job	Trial-3	Train
Completed	customer-churn-predic...	Training job	Trial-2	Train
Completed	customer-churn-predic...	Training job	Trial-1	Train
Completed	customer-churn-predic...	Training job	Trial-0	Train

Mode: Command Ln 1, Col 1 xgboost\_customer\_churn.ipynb

# Introducing Amazon SageMaker Autopilot

## Automatic model creation with full visibility & control

---



### Quick to start

Provide your data in a tabular form & specify target prediction



### Automatic model creation

Get ML models with feature engineering & automatic model tuning automatically done



### Visibility & control

Get notebooks for your models with source code



### Recommendations & Optimization

Get a leaderboard & continue to improve your model

# Build, Train, Deploy Machine Learning Models Quickly at Scale



# AWS DeepRacer improvements

- AWS DeepRacer Evo
  - Stereo camera
  - LIDAR sensor
- New racing opportunities
  - Create your own races
  - Object Detection & Avoidance
  - Head-to-head racing



**Mod your own vehicle**

**Mod specifications**  
The garage shows the DeepRacer vehicles that you can train models for. You can add vehicles by using the "build new vehicle button".

**Sensor modification**  
Swap sensors to improve your DeepRacer's racing performance

**Front-facing camera**  
Single camera that captures the images with sizes of 160 x 120 in front of the agent at 15 fps. The camera has 120 wide angle lens. The images are converted into grey scale before being fed to the neural network.  
» Benefits of the front-facing camera

**Stereo cameras (right/left) sensor**  
Composed of two front-facing cameras, stereo cameras can generate depth information of the objects in front of the agent and thus be used to detect and avoid obstacles on the track. The cameras capture images with the same resolution and frequency. Images from both cameras are converted into grey scale, stacked and then fed into the neural network.  
» Benefits of the stereo camera

**Add-on sensors**

**LIDAR sensor**  
LIDAR is a surveying method that measures a distance to a target by illuminating the target with laser light and measuring the reflected light with a sensor.  
» How LIDAR works with autonomous driving

A diagram of the DeepRacer Evo showing its front view with labels for its sensors: FRONT-FACING CAMERA, STEREO CAMERA LEFT, STEREO CAMERA RIGHT, and LIDAR SENSOR.

# AWS DeepComposer

- The world's first machine learning-enabled musical keyboard
- Compose music using Generative Adversarial Networks (GAN)
- Use a pretrained model, or train your own



# ¡Gracias!

**Memo Doring - @memodoring**

Developer Relations, AWS