SAND2022-6180C

Cloud Frontiers

IoT, Edge, HPC, Quantum, and ML Prototypes and Use Cases in AWS

12 May 2022

John M. Linebarger, PhD, MBA
Sandia National Laboratories
SAND2022-

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Unclassified Unlimited Release

Outline

- Introduction
- IoT vs. Edge Computing
- IoT (Internet of Things)
- Edge Computing
- HPC (High Performance Computing)
- Quantum Computing
- ML (Machine Learning)
- Takeaways
- ML Demo
- Q&A

Introduction

Cloud Frontiers (1)

- In FY2021 I served as the PI (Principal Investigator) of a small project focused on Multicloud Frontiers.
- This category of project explores *existing* technologies that potentially could benefit *multiple* programs at Sandia Labs. They occupy a funding space between single-program applied work and multi-program research work.

Cloud Frontiers (2)

- **The goal of my project as proposed was to explore leading-edge cloud capabilities that all the major cloud vendors were beginning to offer:**
  - Internet of Things (IoT)
  - Edge Computing
  - High Performance Computing (HPC)
  - Quantum Computing
  - Artificial Intelligence (AI) / Machine Learning (ML)
- **Although my title was Multicloud Frontiers, for logistical reasons AWS was almost solely my target cloud platform.**

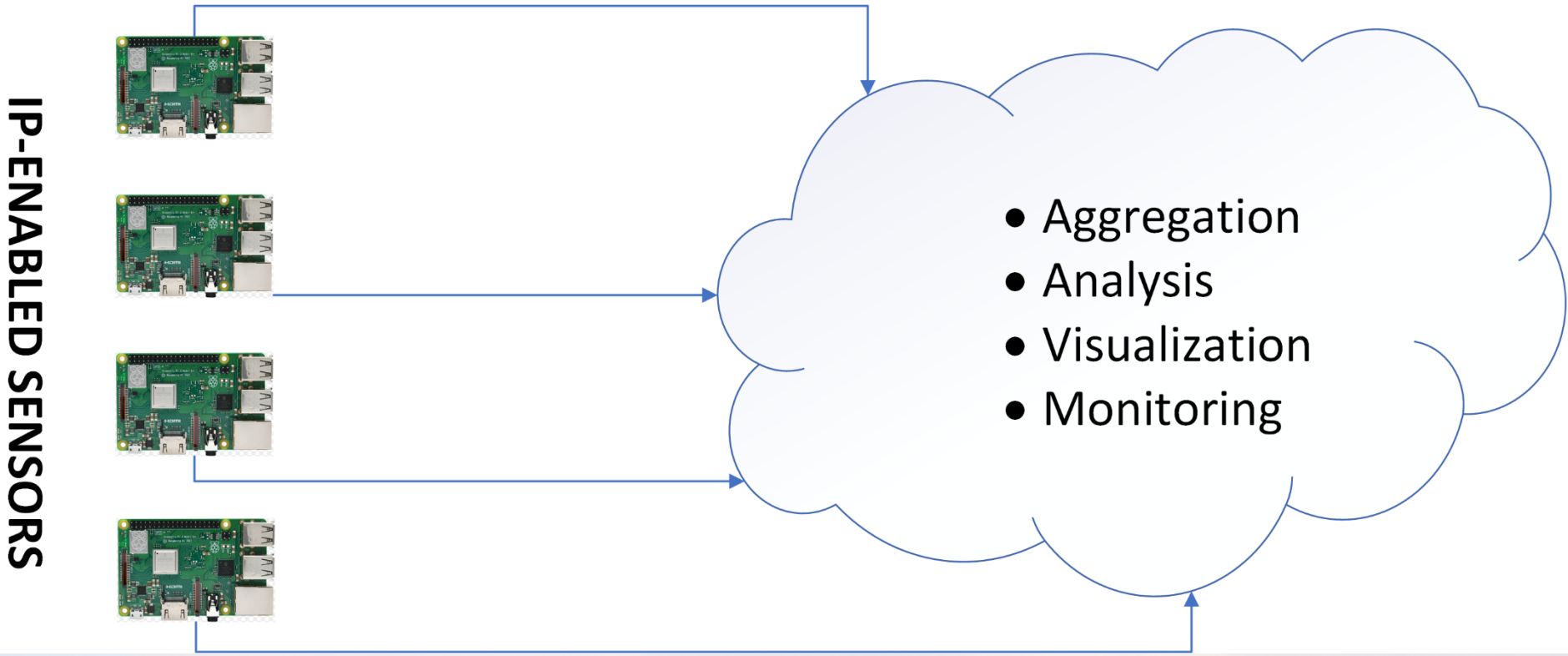
C2E and JWCC contracts are multicloud; DOE has Enterprise Agreement with GCP.

Sandia National Laboratories

IoT vs. Edge Computing

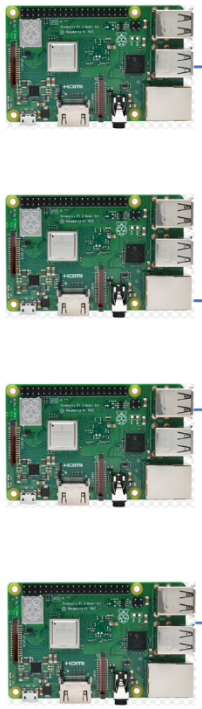
Sandia National Laboratories

IoT Computing Paradigm



Edge Computing Paradigm

IP-ENABLED SENSORS



"The Edge"

- Aggregation
- Analysis
- Filtering

- Visualization
- Monitoring

IoT Computing

Sandia National Laboratories

IoT Computing in AWS

- AWS has a rich set of console-based IoT services as well as device SDKs
- v1.4.9 of the Python IoT SDK was used because it bundles v1.11 of Greengrass Edge Computing SDK instead of v2.x.
- Commercial/GovCloud only

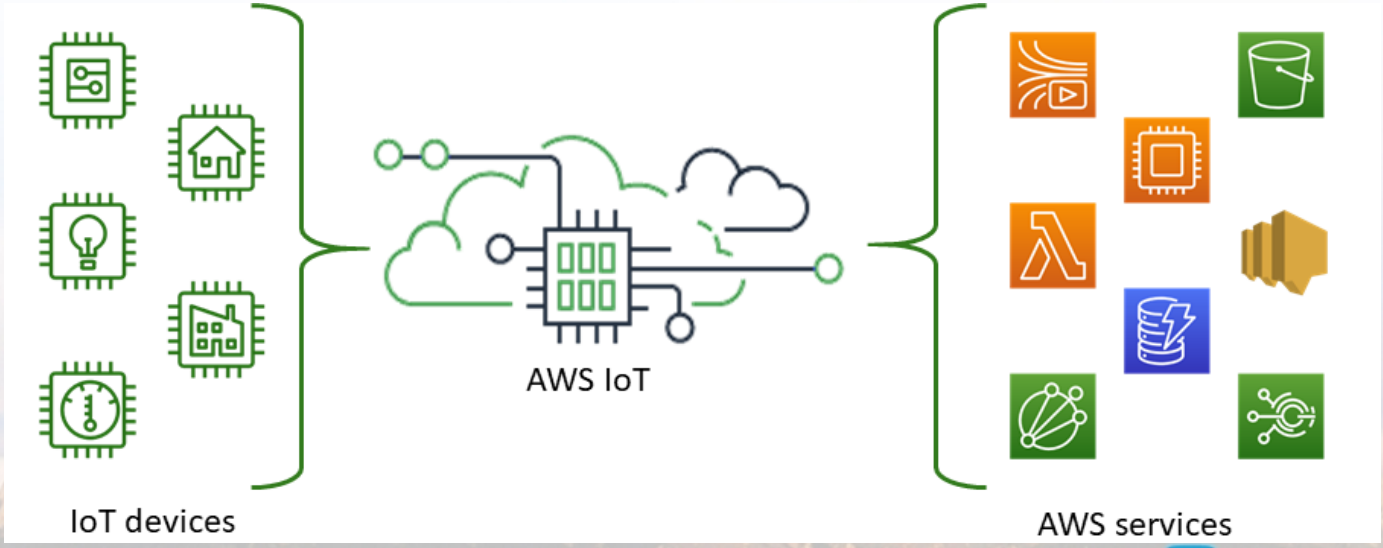


Image Source: https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html

Finding AWS IoT Qualified Devices

AWS Partner Device Catalog

Discover qualified hardware that works with AWS services to help build and deliver successful IoT solutions.

Filter by: Clear all

Search for qualified devices

1-15 of **115** results

Qualifications

- [x] AWS IoT Core
- [ ] AWS IoT Core for LoRaWAN
- [ ] AWS IoT Greengrass
- [ ] FreeRTOS
- [ ] Amazon Kinesis Video Streams

Device Type

- [ ] Asset Tracker
- [ ] Camera
- [ ] Cellular Modem
- [ ] Development Kit
- [ ] Edge Server
- [ ] Gateway / Router
- [ ] Hardware Security Module
- [ ] Industrial PC (IPC)
- [ ] Programmable Automation Controller (PAC)
- [ ] Programmable Logic Controller (PLC)
- [ ] Reference Design
- [ ] RF Module
- [ ] SBC
- [ ] Sensor
- [ ] SOM / COM
- [ ] Starter Kit
- [ ] Storage
- [ ] Other

Industry

Application

Region

Hardware Architecture

Silicon Vendor

Operating System

THE THINGS INDUSTRIES

Gateway/Router

The Things Indoor Gateway

Ultra low-cost LoRaWAN gateway with integrated Wi-Fi backhaul connectivity

Shop now

Qualified for AWS IoT Core

VERIZON.

Asset Tracker

Critical Asset Sensor

Reliable cellular-enabled asset tracking out of the box

Shop now

Qualified for AWS IoT Core

DEVELCO PRODUCTS A/S

Gateway/Router

Squid.link Gateway

Flexible and market-ready white-label IoT gateway solution with wireless connectivity options

Shop now

Qualified for AWS IoT Core

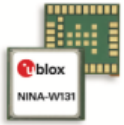
SEGGER MICROCONTROLLER

Development Kit

emPower

SEGGER emPower is an Evaluation Board which includes evaluation software for an easy start to any IoT project.

U-BLOX



RF Module

NINA-W13

NINA-W13 module series with u-connect software for a head start. Secure Wi-Fi connectivity with cloud services natively supported.

QUECTEL WIRELESS SOLUTIONS (上海移远通信技术股份有限公司)

Cellular Modem

BG95

Multi-mode LPWA module supporting LTE Cat M1/Cat NB2/EGPRS and integrated GNSS

Sandia National Laboratories

AWS IoT EduKit

M5Stack Core2 ESP32 IoT Development Kit for AWS IoT EduKit

Brand: M5Stack

2 ratings

Price: $42.00 & FREE Returns

Get $50 off instantly: Pay $0.00 ~~$42.00~~ upon approval for the Amazon Rewards Visa Card. No annual fee.

- Reference hardware kit for use with AWS IoT EduKit;
- ESP32-D0WDQ6-V3, supports 2.4GHz WiFi, Bluetooth 4.2, BLE; 16M Flash, 8M PSRAM; Built-in ATECC608 Trust&GO hardware encryption chip;
- Capacitive touch screen; Built-in PDM microphone, power indicator, 6-Axis IMU, vibration motor, I2S codec, Amplifier, Speaker, RTC, power button, reset button, 10 x RGB LEDs; SD card slot (supports up to 16GB); Built-in 500mAh lithium-ion battery, equipped with a power management chip;
- Supports FreeRTOS, MicroPython, UIFlow, Ar duino development frameworks;
- Validated through AWS Device Qualification Program

5% off coupon

LABISTS Raspberry Pi 4 8GB RAM Starter Kit with 128GB Micro SD Card (8GB RAM)

269

$149.99 prime

Sponsored

$42.00

& FREE Returns

FREE delivery: **Sunday, March 28** Details

Fastest delivery: **Thursday, March 25**

Order within 6 hrs and 31 mins

Details

In Stock.

Qty: 1

Add to Cart

Buy Now

Secure transaction

Ships from Amazon

Sold by M5Stack Official Store

prime

Enjoy fast, FREE delivery, exclusive deals and award-winning movies & TV shows with Prime

Try Prime and start saving today with **Fast, FREE Delivery**

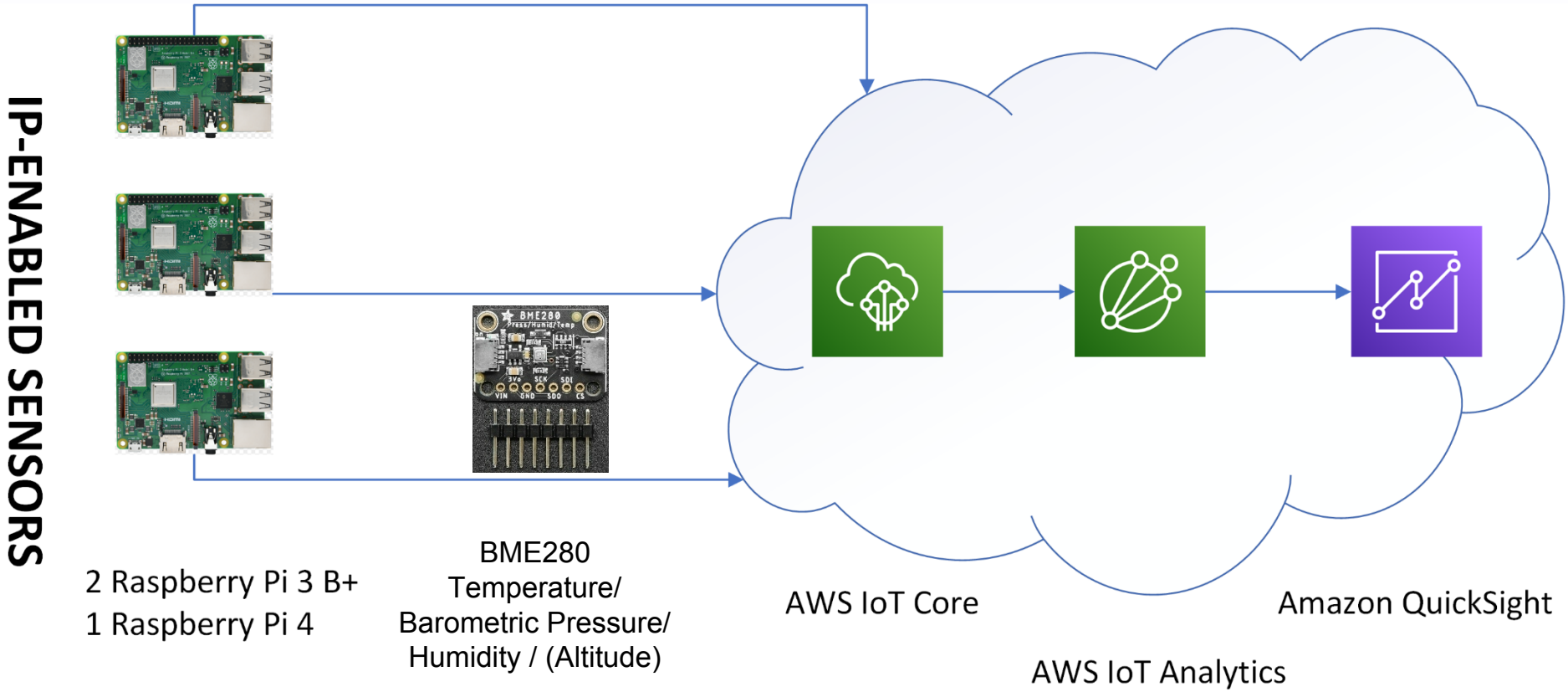
https://www.amazon.com/dp/B08VGRZYJR/

AWS IoT Computing Architecture

- MQTT (Message Queuing Telemetry Transport) pub/sub messages used for device communication since supports low bandwidth/high latency environments
- Each IoT device (called a Thing) has a Shadow that the cloud can populate with a Desired value and the device can populate with a Reported value using MQTT and a REST API endpoint
- Things can be typed and grouped
- Device-specific PKI certs and associated IAM policies used for security

Sandia National Laboratories

IoT Prototype Architecture



IoT Computing Prototype

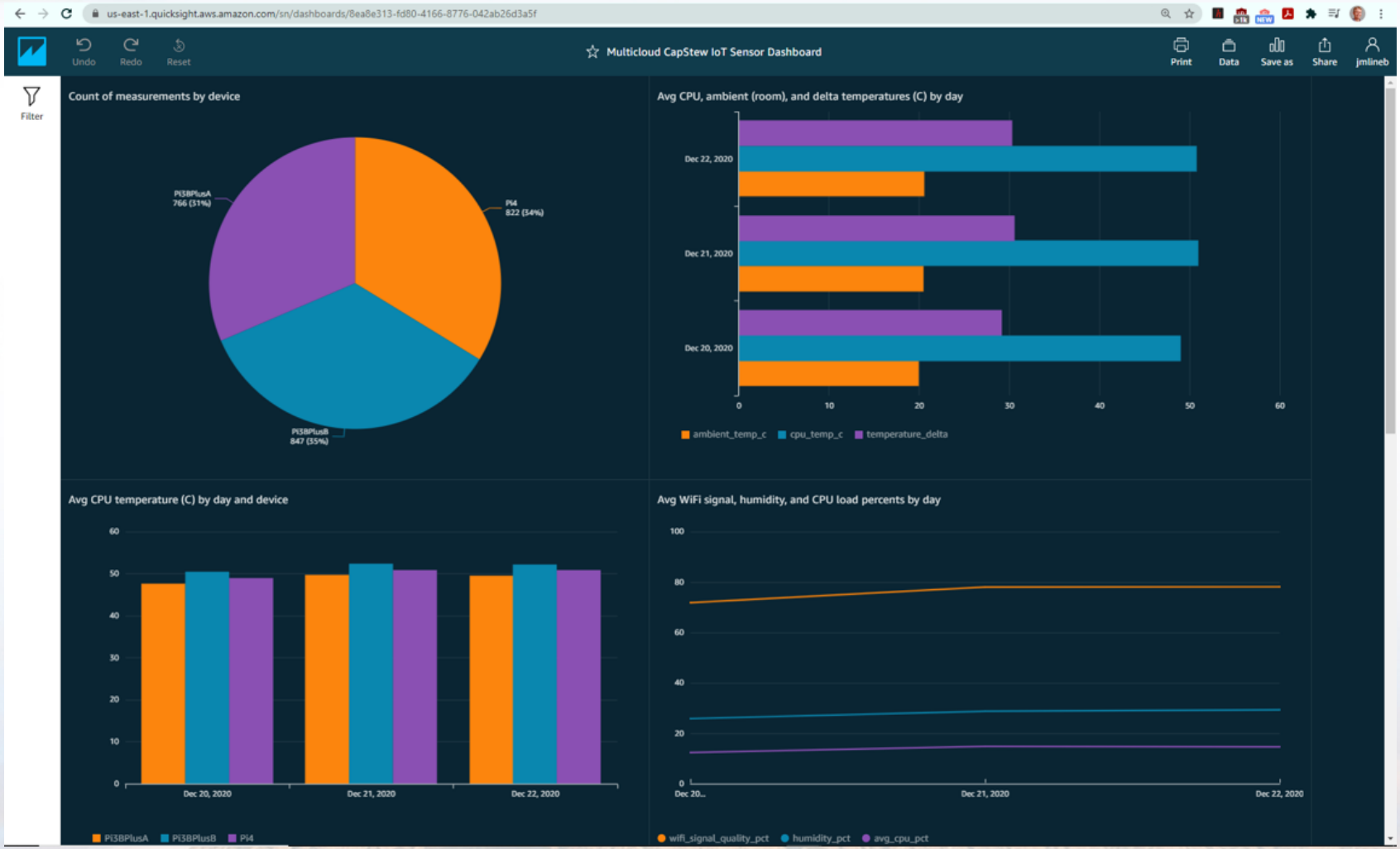
- 3 Raspberry Pi devices with GPIO/ I2C sensors sent data to IoT Core
- IoT Analytics created data table (dataset) from the data pushed by IoT Core
- QuickSight pulled and visualized that dataset

- Data from Raspberry Pi:
  - CPU temperature
  - Average CPU percent
  - Number of processes
  - WiFi signal quality
- Data from GPIO/ I2C sensor:
  - Ambient temperature
  - Ambient humidity
  - Ambient barometric pressure
  - Altitude (calculated)

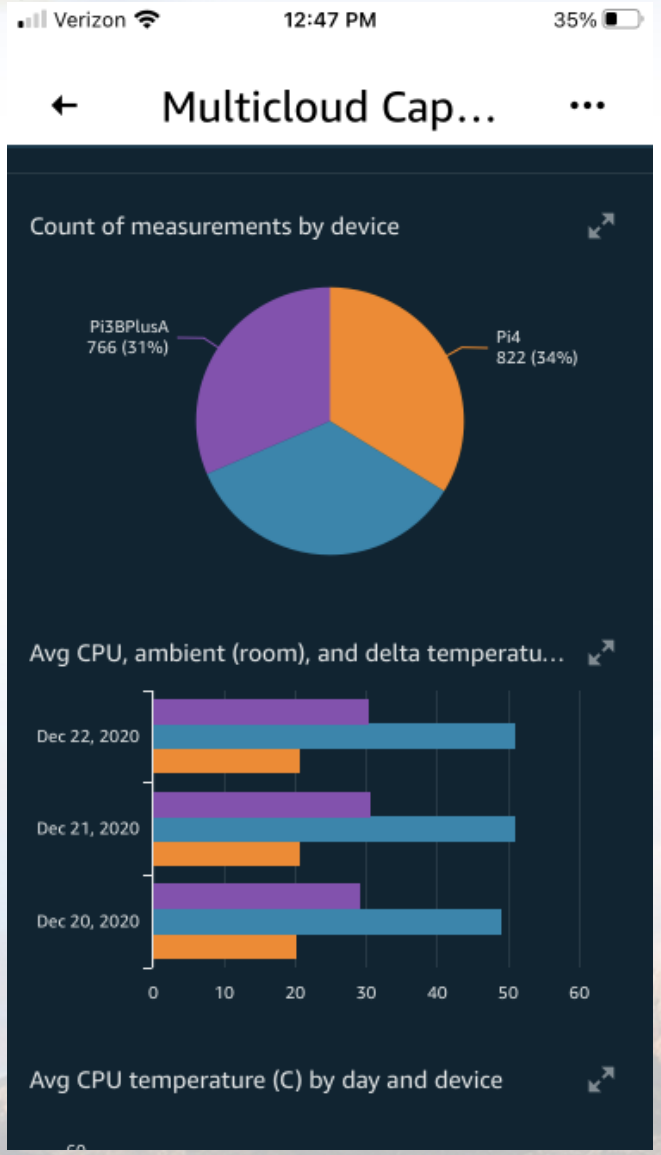
Sandia National Laboratories

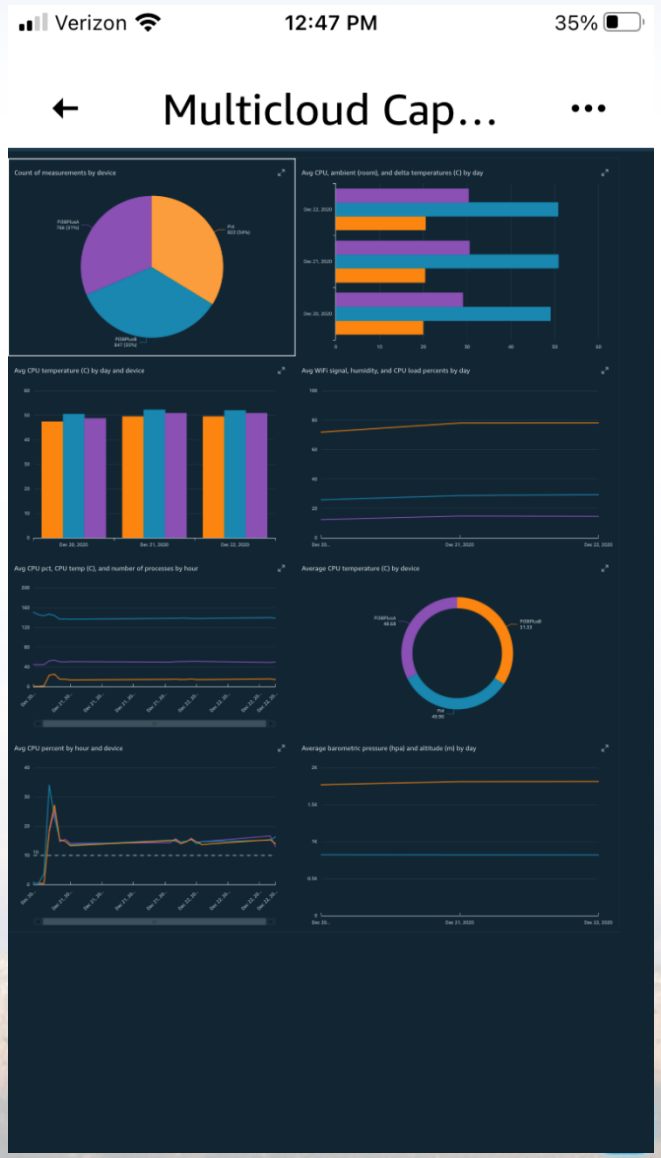
R Lab

Results: QuickSight Dashboard

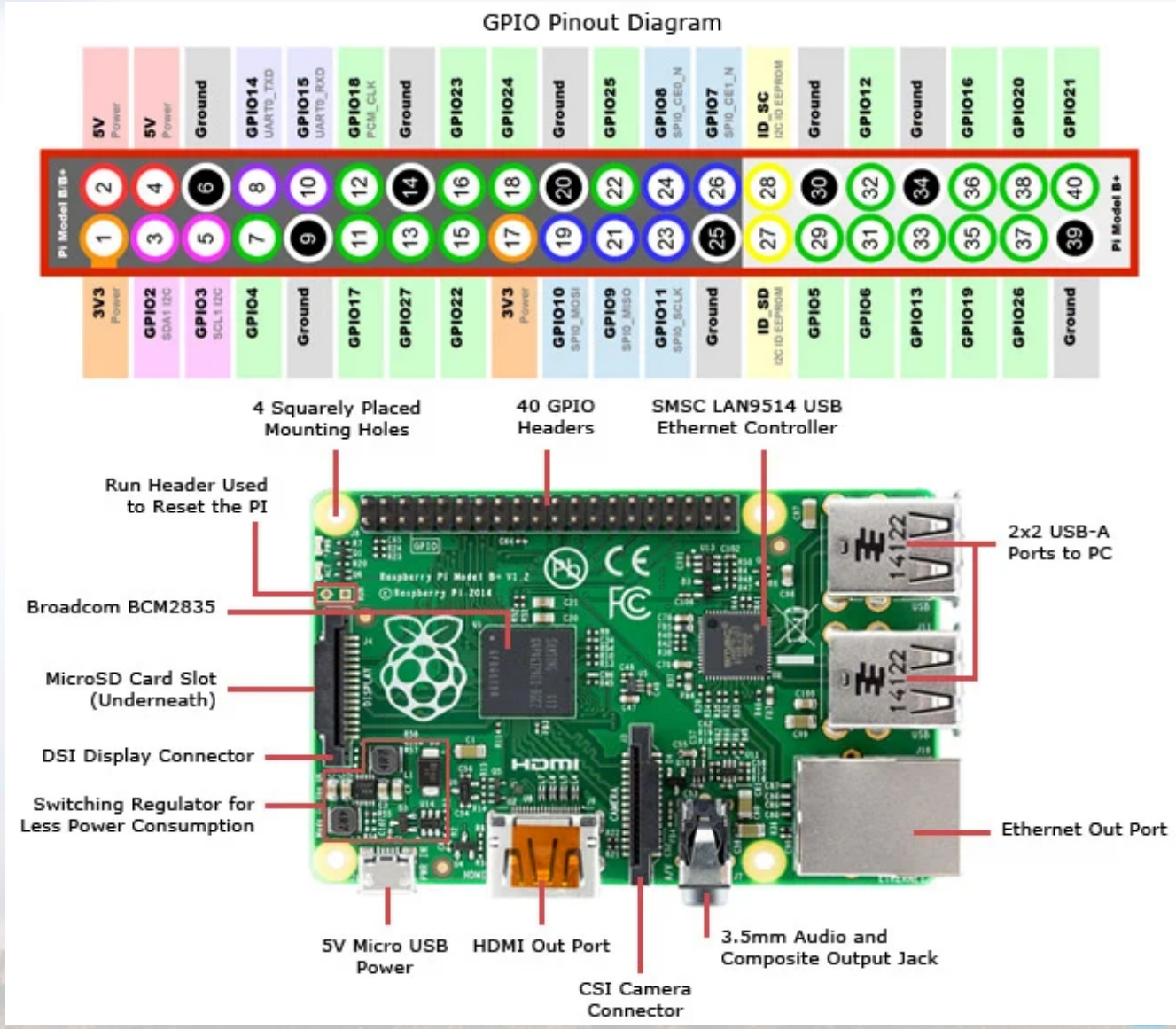


Results: QuickSight Mobile App

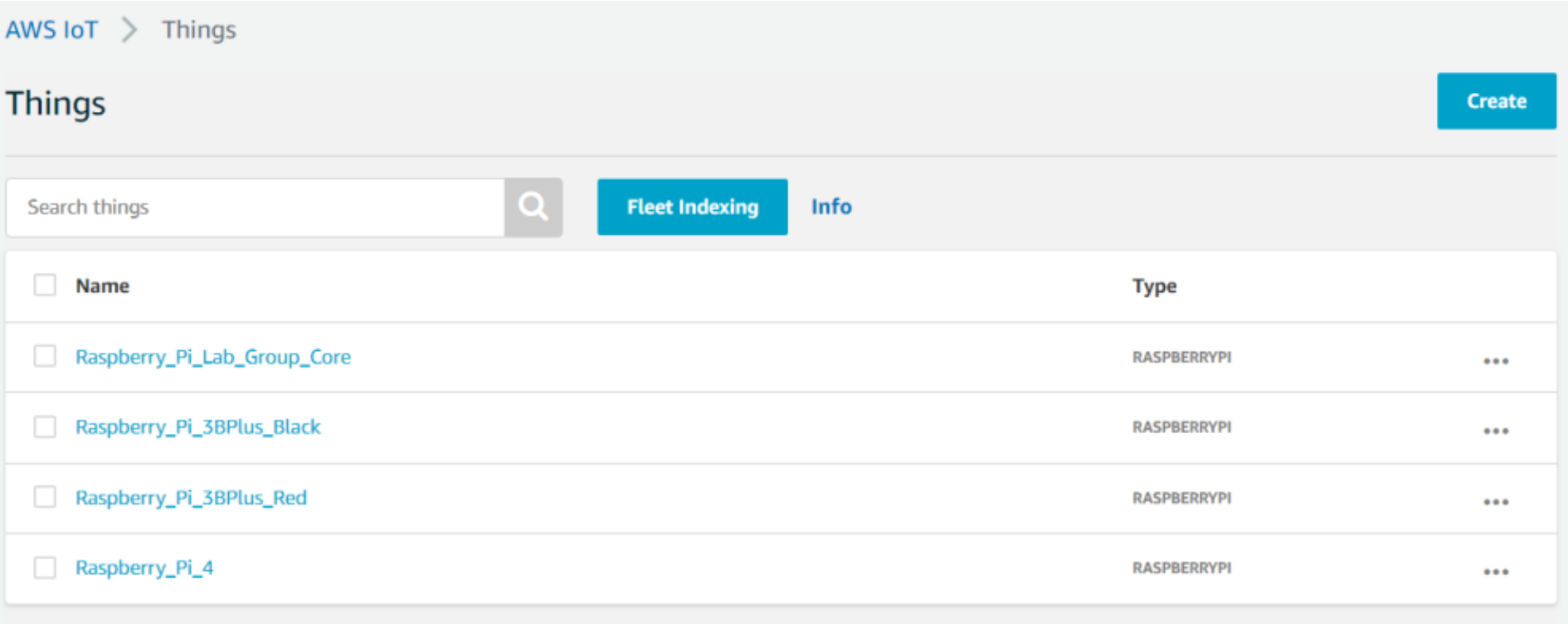




Mechanics: Raspberry Pi Sensor I/F

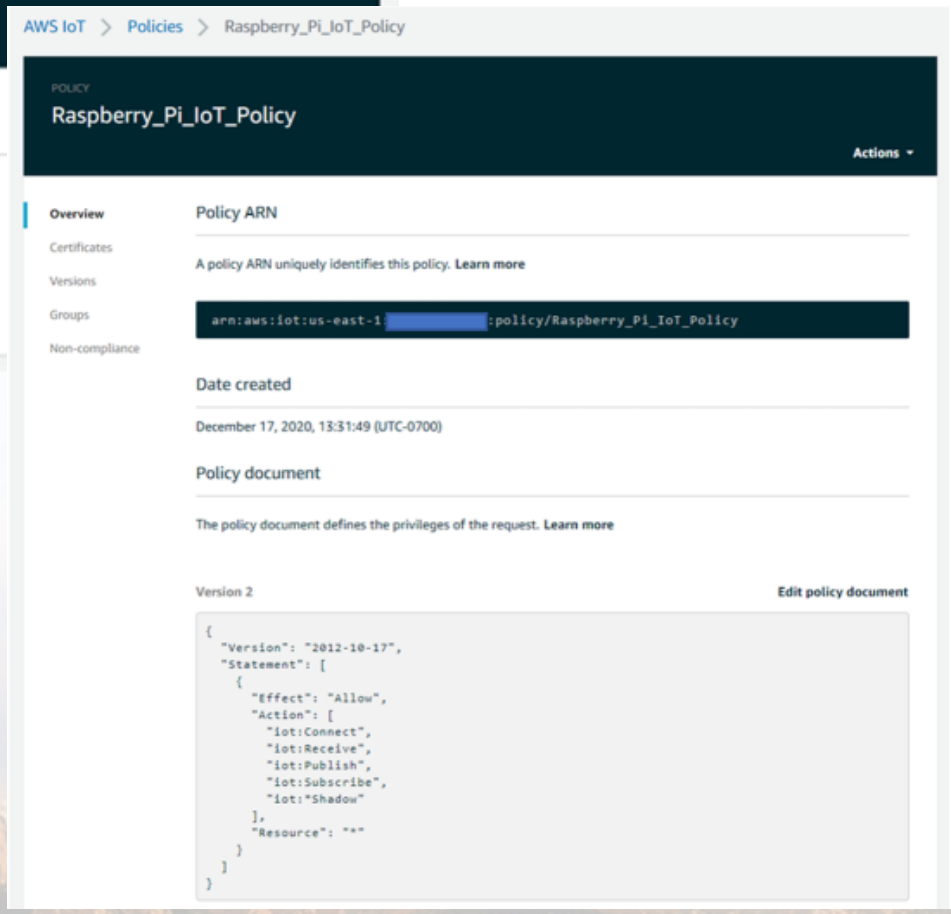


Mechanics: Things (IoT Devices)



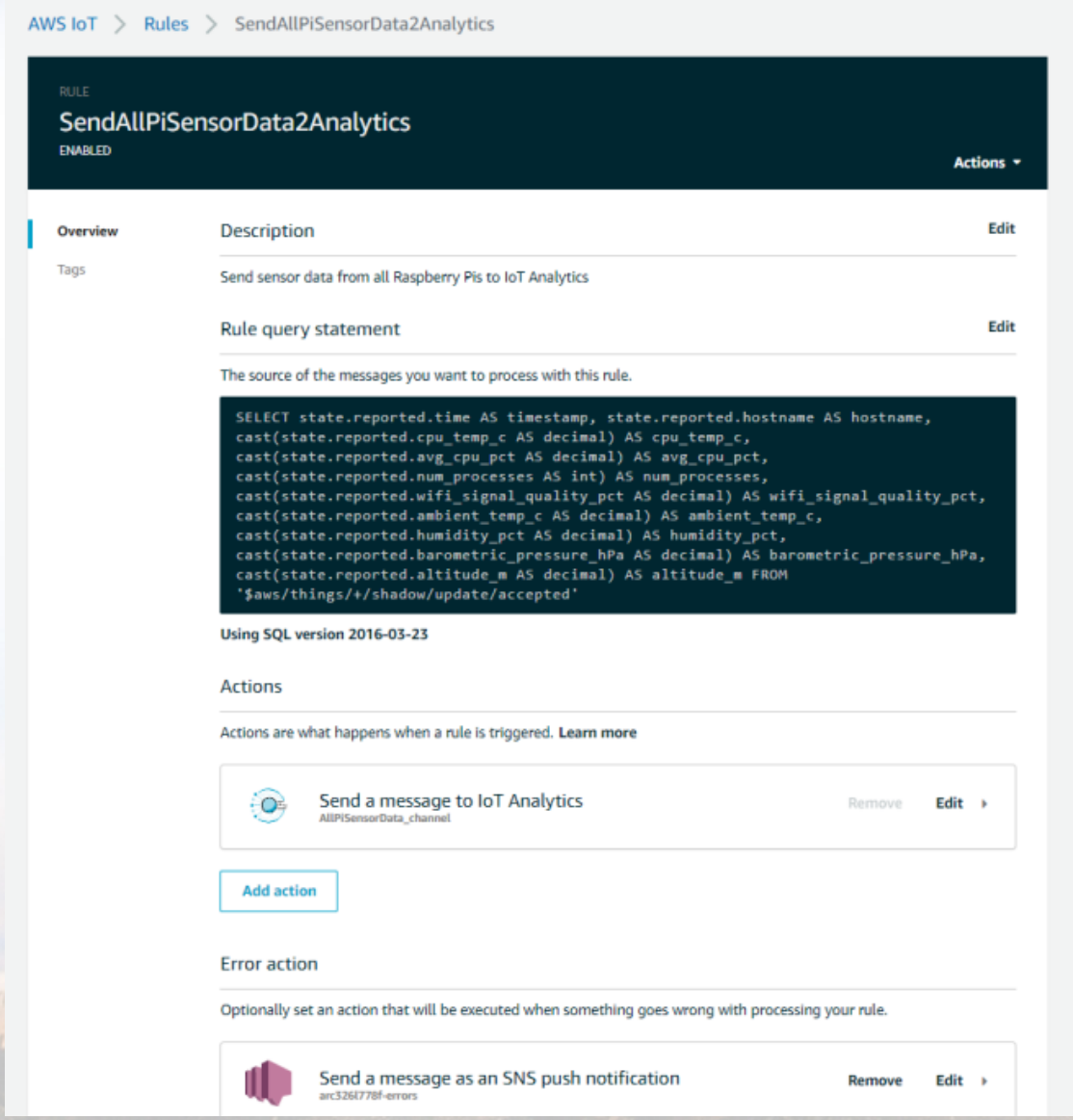
Sandia National Laboratories

Mechanics: PKI Certs with IAM Policies



Sandia National Laboratories

Mechanics: Auto Push to IoT Analytics



Mechanics: scikit-learn Forecasting

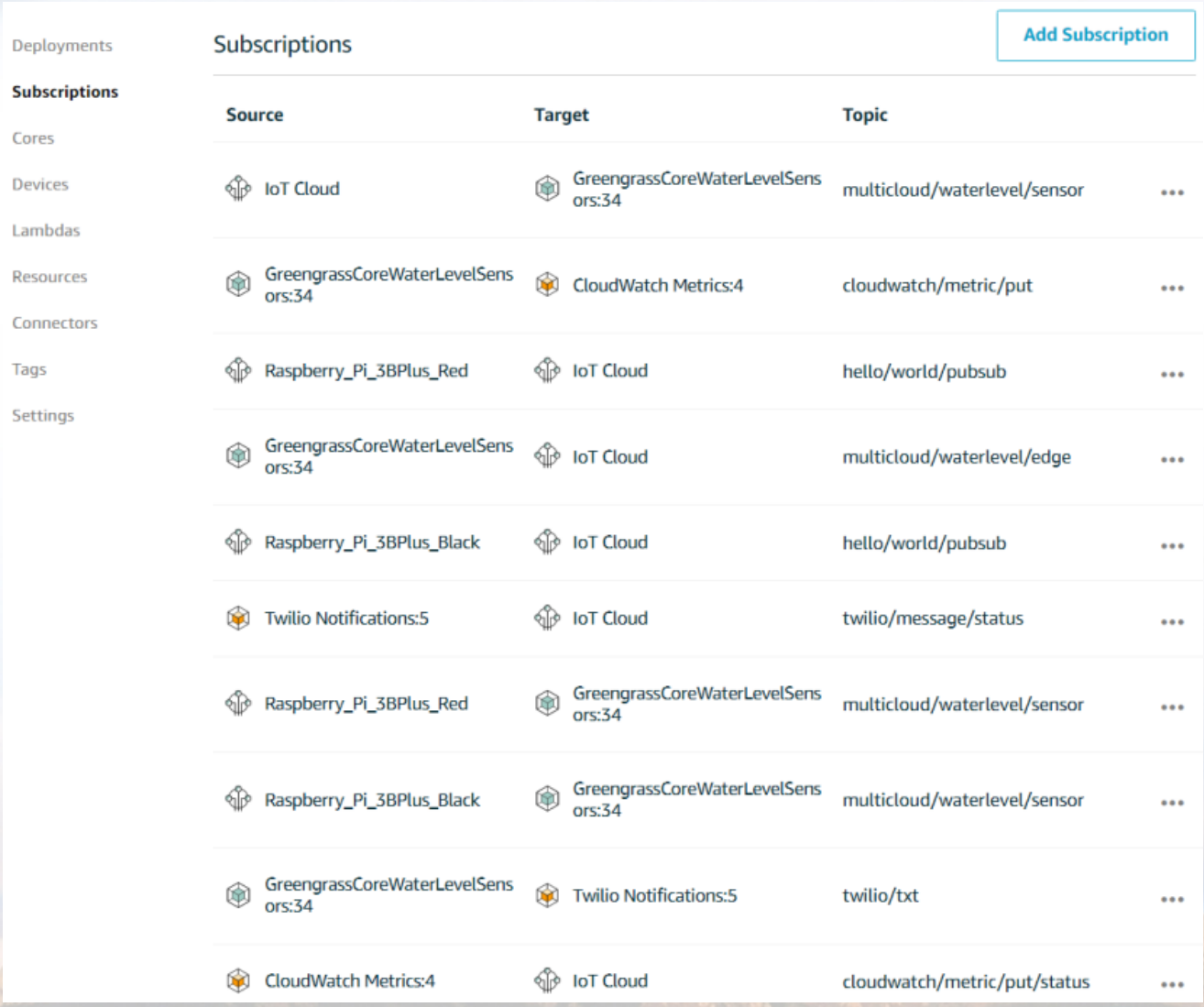
```
289 def extrapolate_minutes_to_exceed_threshold():
290     """
291     Run linear regression to extrapolate minutes to exceed threshold in the near future.
292     Note that scikit-learn does not seem to break if the file contains only a small number of data points.
293     """
294     minutes_to_exceed_threshold = 0
295     affected_sensor = ''
296     for sensor in sensor_data.keys():
297         # Run linear regression on the sensor data file (i.e., train with the *entire* data set instead of a partitioned training data set)
298         dataset = pd.read_csv(sensor_data[sensor]['filename'])
299         X = dataset.iloc[:, :-1].values
300         Y = dataset.iloc[:, 1].values
301         regressor = LinearRegression()
302         regressor.fit(X, Y)
303         # Project into the near-term future; first find the highest epoch seconds in the sensor data file.
304         highest_epoch_seconds = 0
305         with open(sensor_data[sensor]['filename'], 'r') as file:
306             reader = csv.reader(file, delimiter = ',')
307             index = -1
308             for row in reader:
309                 print(row)
310                 index += 1
311                 if index == 0:
312                     continue
313                 if int(row[0]) > highest_epoch_seconds:
314                     highest_epoch_seconds = int(row[0])
315         # Now project (extrapolate) into the near-term future based on the regression line calculated
316         # (i.e., test with extrapolated values instead of a dedicated test data set)
317         future_epoch_seconds = highest_epoch_seconds + 60
318         for x in range(int(future_projection_minutes)):
319             future_ext=[[future_epoch_seconds]]
320             y_ext = regressor.predict(future_ext)
321             if (y_ext >= float(threshold_pct)):
322                 if minutes_to_exceed_threshold == 0 or minutes_to_exceed_threshold > x + 1:
323                     minutes_to_exceed_threshold = x + 1
324                     affected_sensor = sensor
325                 break
326             future_epoch_seconds += 60
327     return minutes_to_exceed_threshold, affected_sensor
```

Regression (fit model to recent historical data)

Prediction (near-term extrapolation)

Sandia National Laboratories

Mechanics: Subscription Table



Mechanics: Updating to New Version

- **On Lambda dashboard:**
  - Change code and save it
  - Deploy new code and create new version
- **On IoT Core dashboard:**
  - Remove old version of Lambda function
  - Add and parameterize new version of Lambda function (including all environment variables)
  - Delete subscriptions for old version
  - Re-create subscriptions for new version
  - Reset deployment and deploy
- **Elapsed time: 10-20 minutes; error-prone**
- **Mitigations**
  - Modified code to (mostly) run locally to debug syntax errors
  - Used Jupyter Notebook for prototyping

Sandia National Laboratories

Mechanics: Jupyter Notebooks

jupyter scikit-learn Last Checkpoint: a minute ago (autosaved) Logout

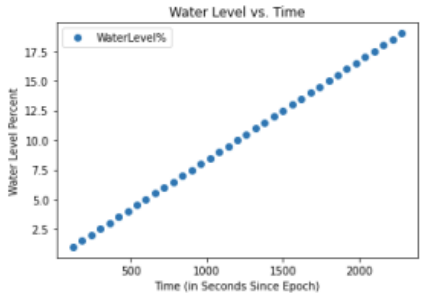
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
    file_handle.write(file_line)
file_handle.close()
```

Perform linear regression (i.e., supervised learning) on the latest version of the file and plot it

In [25]:
```
wdataset = pd.read_csv('D:\MulticloudFrontiers\datasets\water_levels.csv')
wX = wdataset.iloc[:, :-1].values
wY = wdataset.iloc[:, 1].values
from sklearn.linear_model import LinearRegression
wregressor = LinearRegression()
wregressor.fit(wX, wY)
# Plot the file data
wdataset.plot(x='EpochSeconds', y='WaterLevel%', style='o')
plt.title('Water Level vs. Time')
plt.xlabel('Time (in Seconds Since Epoch)')
plt.ylabel('Water Level Percent')
plt.show()
```



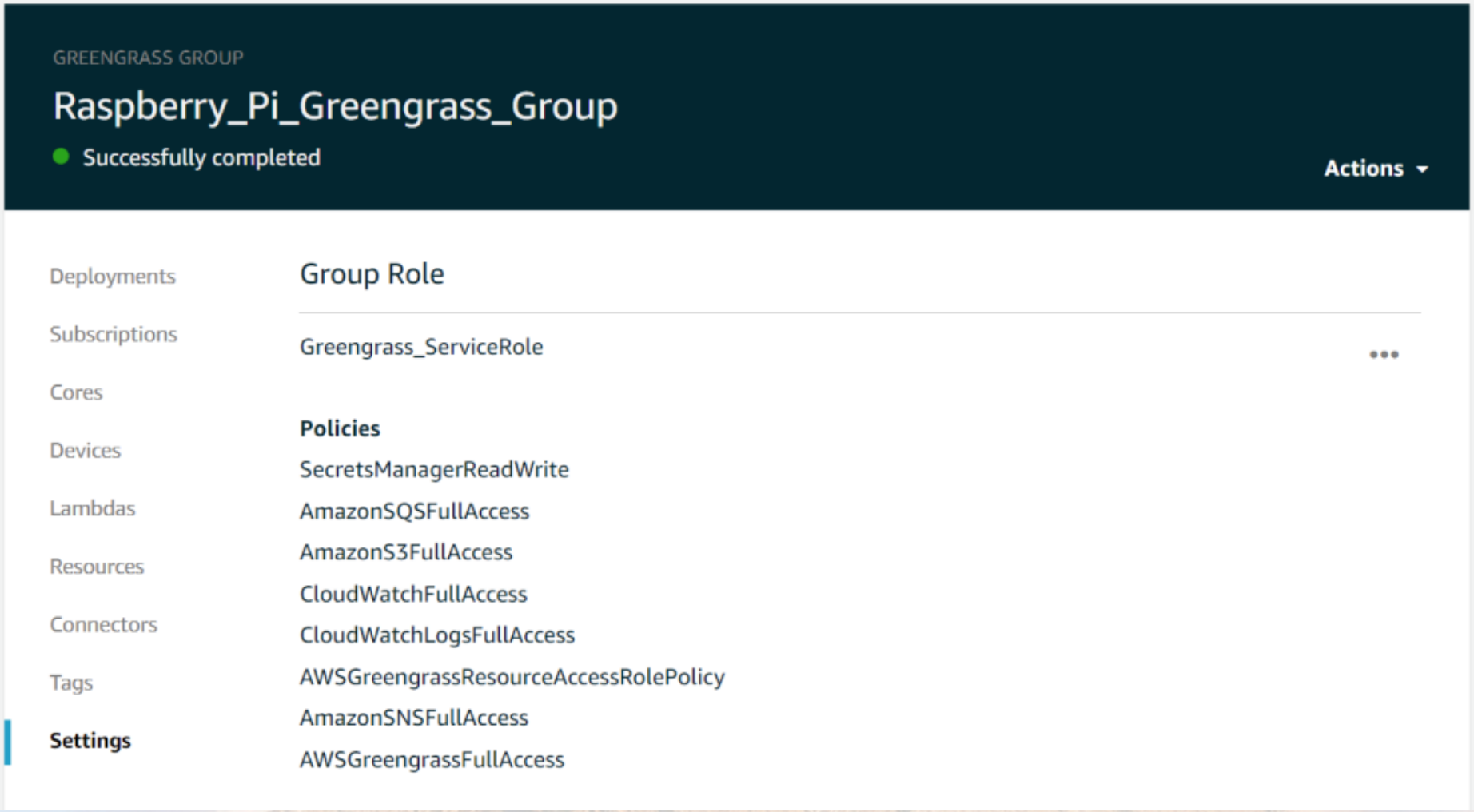
Predict (forecast) every minute for an hour into the future. Greate way to alert to a slow leak.

In [26]:
```
THRESHOLD = 25.0
future_epoch = highest_epoch + 60
for x in range(60):
    future_ext=[[future_epoch]]
    y_ext = wregressor.predict(future_ext)
    # numpy evidently wants the old-style format strings
    print("Future epoch: %d, future water level: %1.1f" % (future_epoch, y_ext))
    if (y_ext >= THRESHOLD):
        minutes = x+1
        print ("Water level threshold of %1.1f percent is projected to be breached %1.1f minutes into the future." % (THRESHOLD,m
        break
    # print("{:.1f}".format(y_ext))
    future_epoch += 60
```

```
Future epoch: 2340, future water level: 19.5
Future epoch: 2400, future water level: 20.0
Future epoch: 2460, future water level: 20.5
```

Sandia National Laboratories

Mechanics: Group IAM Role / Policies



High Performance Computing (HPC)

Sandia National Laboratories

HPC on AWS (1)

- **AWS provides an open source cluster management tool called AWS ParallelCluster. It stands up an HPC cluster using a configuration file and pcluster CLI commands.**
- **Under the covers, AWS ParallelCluster creates and instantiates a set of nested CloudFormation templates.**
  - This is important to know because if something goes wrong in creating or deleting the cluster, the only way to delete a corrupted cluster is at the CloudFormation level.
  - I have received many “Security Token Expired” or network errors trying to create or delete clusters.

Sandia National Laboratories

HPC on AWS (2)

This is a significant behavior (and performance) difference from a dedicated HPC computer, where the nodes are always running

- **What standing up a cluster actually does is to create a master node (EC2 instance) that is configured for a particular HPC job scheduler, which will stand up compute nodes when a job is submitted.**
- **AWS reduced support to two HPC job schedulers at the end of 2021.**
  - Slurm (popular; dynamically spins compute nodes up/down)
  - AWS Batch
- **AWS has replicated the traditional batch-oriented, command-driven interface for HPC. If it ain't broke …**

Sandia National Laboratories

HPC Example: Calculating PI (1)

- **First, configure pcluster itself (one time).**
  - `$env:AWS_PROFILE="commercial"` [if SRN PowerShell]
  - `pcluster configure`
- **Next, stand up a cluster (~10-12 minutes)**
  - `$env:AWS_PROFILE="commercial"`
  - `pcluster create PI --config PiDemoCluster.cfg`
- **Example output from cluster creation**
  - `Beginning cluster creation for cluster: PI`
  - `Creating stack named: parallelcluster-PI`
  - `Status: parallelcluster-PI - CREATE_COMPLETE`
  - `MasterPublicIP: <public_IP>`
  - `ClusterUser: ec2-user`
  - `MasterPrivateIP: <private_IP>`

Sandia National Laboratories

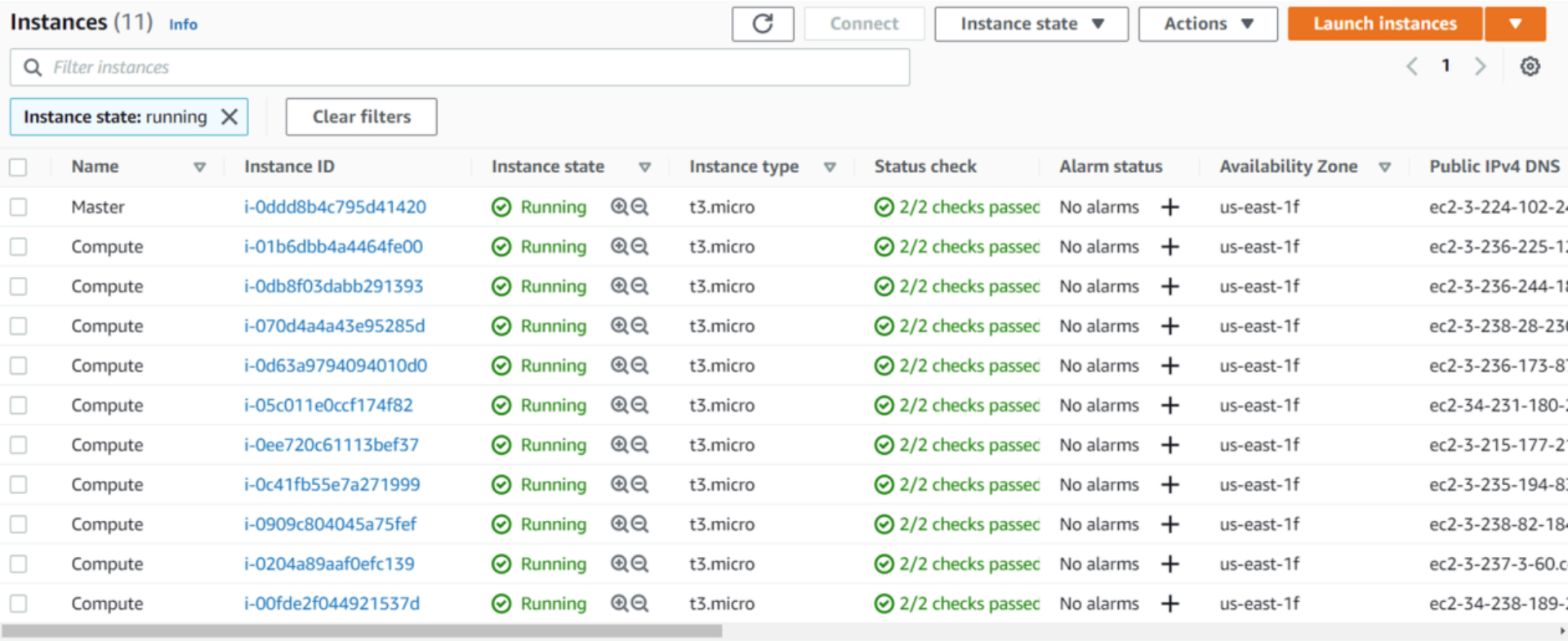
HPC Example: Calculating PI (2)

- **Now SSH into master node using Public IP address and configured PKI key**
  - `pcluster ssh PI -i <*.ppm key with restricted permissions>`
  - (Use PuTTY with *.ppk key in Windows.)
- **Download code and scripts from S3**
- **Compile on master and run on cluster (~3-5 minutes)**
  - `mpicc MPICalculatePI.c -o MPICalculatePI`
  - `sbatch ./MPICalculatePI.sh`
  - `{squeue | sinfo}`

In MPI, one of the compute nodes is considered the root node

Sandia National Laboratories

Master Node and Compute Nodes



Instances (11) Info

Connect | Instance state | Actions | Launch instances

Filter instances

1

Instance state: running | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Master	i-0ddd8b4c795d41420	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-3-224-102-2
Compute	i-01b6dbb4a4464fe00	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-3-236-225-1
Compute	i-0db8f03dabb291393	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-3-236-244-1
Compute	i-070d4a4a43e95285d	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-3-238-28-23
Compute	i-0d63a9794094010d0	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-3-236-173-8
Compute	i-05c011e0ccf174f82	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-34-231-180-
Compute	i-0ee720c61113bef37	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-3-215-177-2
Compute	i-0c41fb55e7a271999	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-3-235-194-8
Compute	i-0909c804045a75fef	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-3-238-82-18
Compute	i-0204a89aaf0efc139	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-3-237-3-60.c
Compute	i-00fde2f044921537d	Running	t3.micro	2/2 checks passec	No alarms	us-east-1f	ec2-34-238-189-

Sandia National Laboratories

HPC Example: Calculating PI (3)

- **Check output in *.out and *.err files**
  - `Number of nodes was 12; pi calculated as approximately 3.1421713566497971; error from the PI25DT constant is 0.0005787030600040.`

Never was able to successfully request spot instances

- **Slurm will spin down compute nodes automatically after a period of non-use**
- **Other useful pcluster commands**
  - `pcluster list`
  - `pcluster status PI`
  - `pcluster instances PI`
  - `pcluster {stop | start | update}`
- **Delete the cluster**
  - `pcluster delete PI`

Sandia National Laboratories

Running a CFD Simulation on AWS

- **A more typical HPC workload is a Computational Fluid Dynamics (CFD) simulation**
- **OpenFOAM was chosen because it is open source**
- **OpenFOAM was both compiled and executed in HPC mode**
  - Compiled with 36 nodes on 1 c5n.18xlarge EFA instance
  - Executed with 108 nodes on 3 c5n.18xlarge EFA instances
- **The data plane for the nodes was a shared Lustre-based parallel file system**
- **Visualized the output remotely and locally**

I also experimented with the FDS-SMV fire dynamics simulator

Amazon FSx for Lustre

Sandia National Laboratories

OpenFOAM CFD Simulation Flow

Takes ~15 minutes

- **Stand up the CFD cluster**
  - `pcluster create CFD -c ./CFD.cfg`
- **SSH to master node of cluster**
  - `pcluster ssh CFD -i ./<key_pair>.pem`
- **Change directory to FSx shared file system; download OpenFOAM from S3.**

Takes well over an hour

- **From master node, compile OpenFOAM with 36 EFA-enabled compute nodes**
  - `sbatch ./foam_compile.sh`

Takes ~3-5 minutes

- **From master node, execute OpenFOAM motorBikeDemo (108 EFA compute nodes)**
  - `sbatch ./foam_submit.sh`

Sandia National Laboratories

pcluster / slurm for CFD Prototype

```
PS C:\Users\John\Multicloud> ls

    Directory: C:\Users\John\Multicloud

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---            6/8/2021  2:33 PM           1679 cfd_lab_key_pair.pem
-a---           6/12/2021  7:12 PM           1296 config
-a---           6/12/2021  7:12 PM           1238 config-arm

PS C:\Users\John\Multicloud> pcluster create cfd -c .\config
Beginning cluster creation for cluster: cfd
Info: There is a newer version 2.10.4 of AWS ParallelCluster available.
Creating stack named: parallelcluster-cfd
Status: parallelcluster-cfd - CREATE_COMPLETE
ClusterUser: ec2-user
MasterPrivateIP: 10.0.0.88
```

```
[ec2-user@ip-10-0-0-88 etc]$ cat compile.sh
#!/bin/bash
#SBATCH --job-name=foam-36
#SBATCH --ntasks=36
#SBATCH --output=%x_%j.out
#SBATCH --partition=compute
#SBATCH --constraint=c5n.18xlarge
module load openmpi
source /fsx/OpenFOAM/OpenFOAM-v2012/etc/bashrc
export WM_NCOMPPROCS=36
cd /fsx/OpenFOAM/OpenFOAM-v2012/
./Allwmake > log.dat
```

Sandia National Laboratories

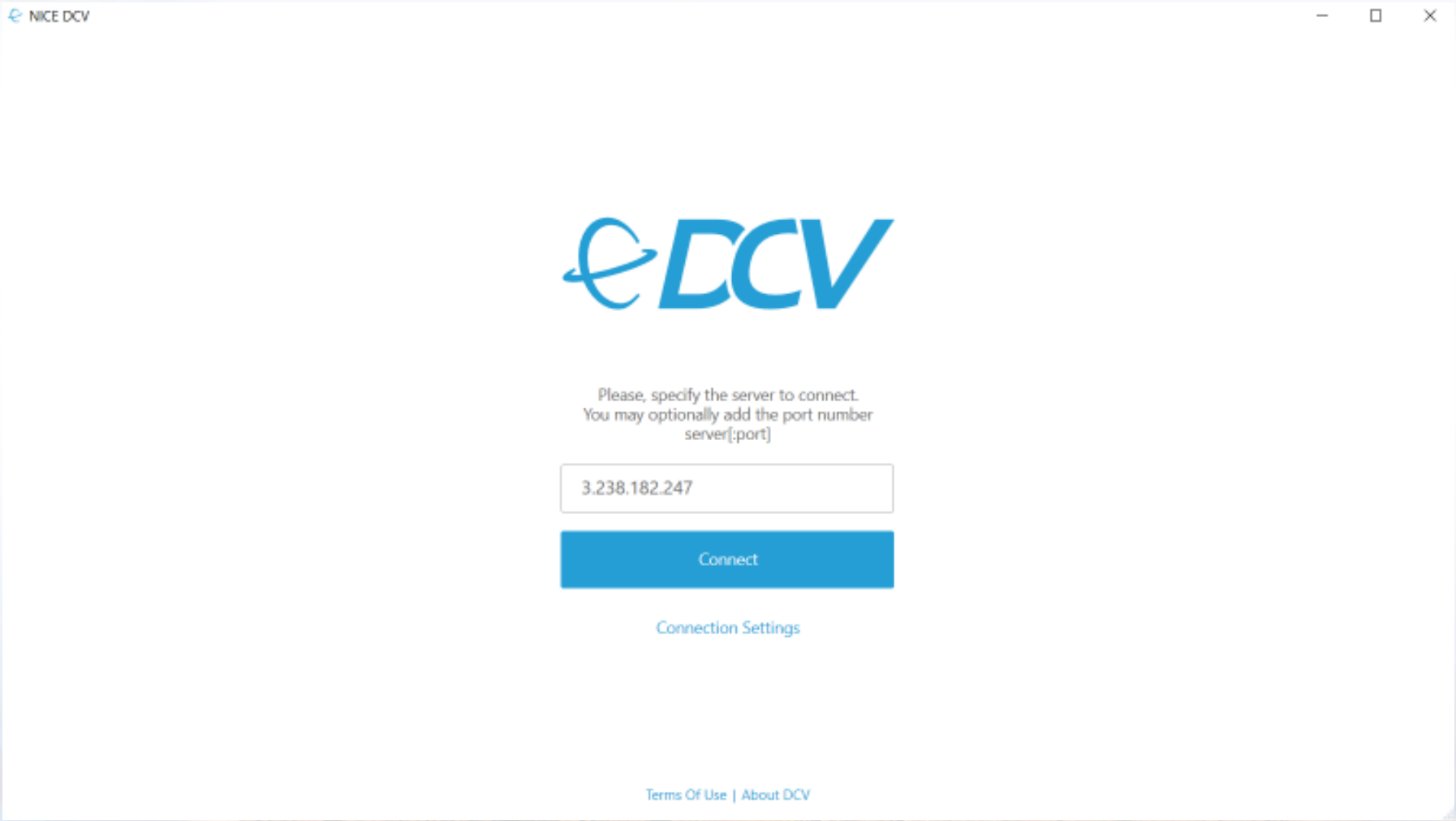
Remote Visualization with NICE DCV

- Stand up a GPU-based visualization instance using a NICE DCV AMI
- Attach to the same FSx for Lustre parallel file system as the CFD cluster
- Configure DCV permissions and password and start DCV server
- Install DCV client on local machine
- Log in to visualization instance from local DCV client
- Invoke remote visualization software, which ships pixels to the local DCV client

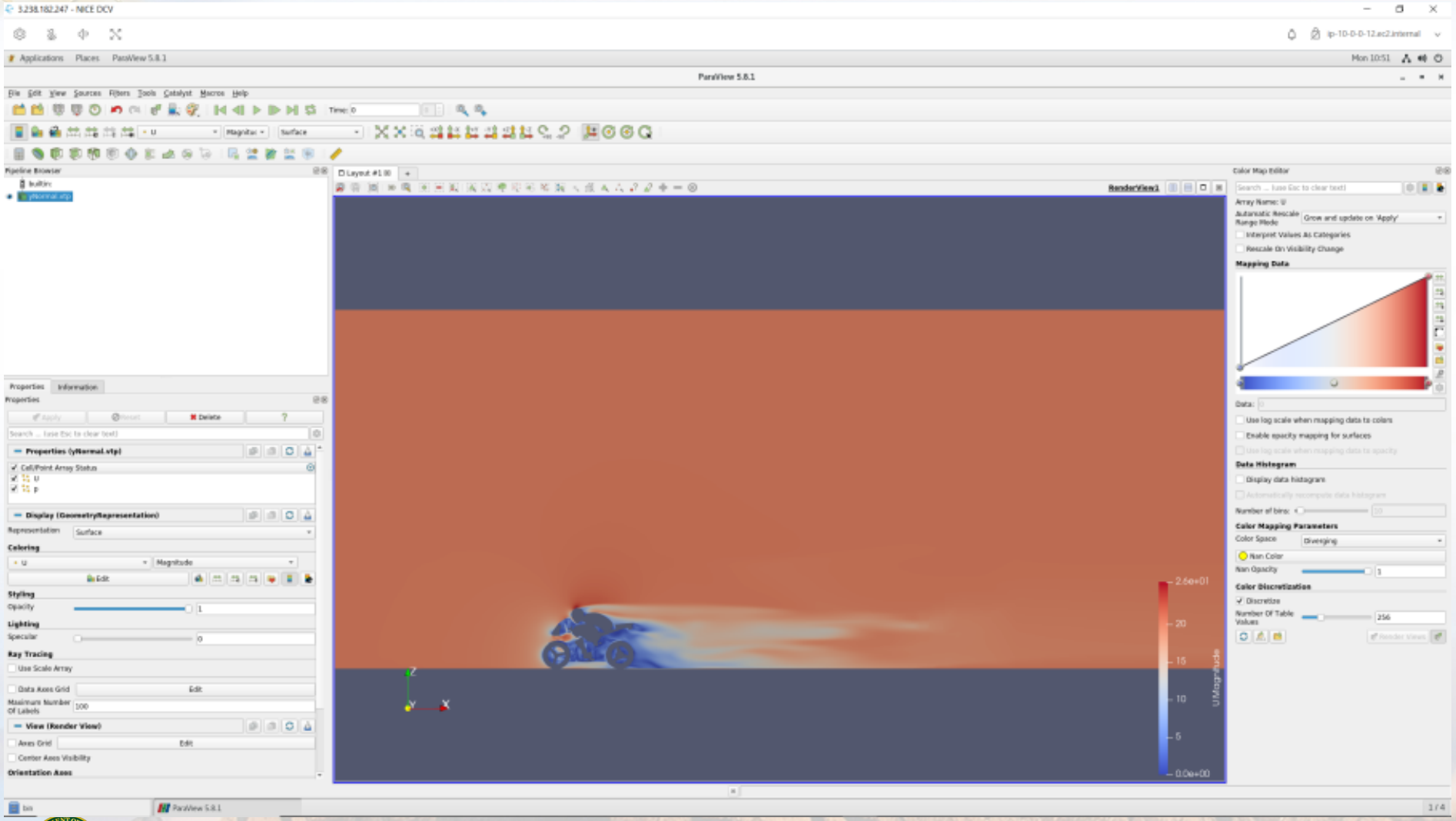
Will not work through SRN proxy

Sandia National Laboratories

Client Login to Visualization Instance



Remote NICE DCV Viz with ParaView



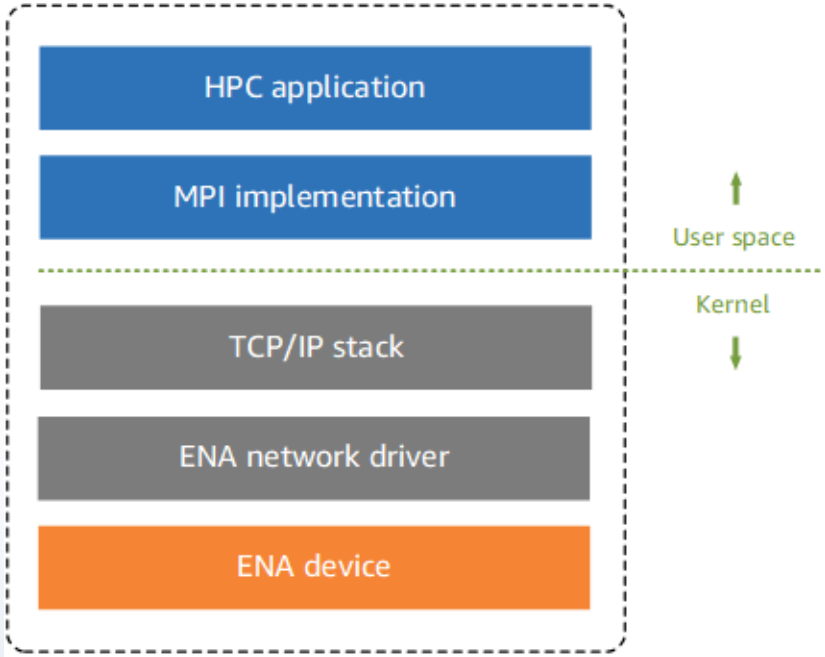
Elastic Fabric Adapter (EFA)

- **EFA is AWS's high-throughput OS bypass networking capability used for demanding HPC jobs**
  - EFA is AWS's equivalent of InfiniBand in traditional HPC
  - The libfabric API is used with a plugin for AWS's reliable UDP protocol called SRD (Scalable Reliable Datagram)
- **Available in a few high end 100 Gbps network bandwidth instances (such as c5n.18xlarge) which are put in the same placement group in the same AZ**
- **Depends on AWS's proprietary Nitro hypervisor, which is mostly HW-based**

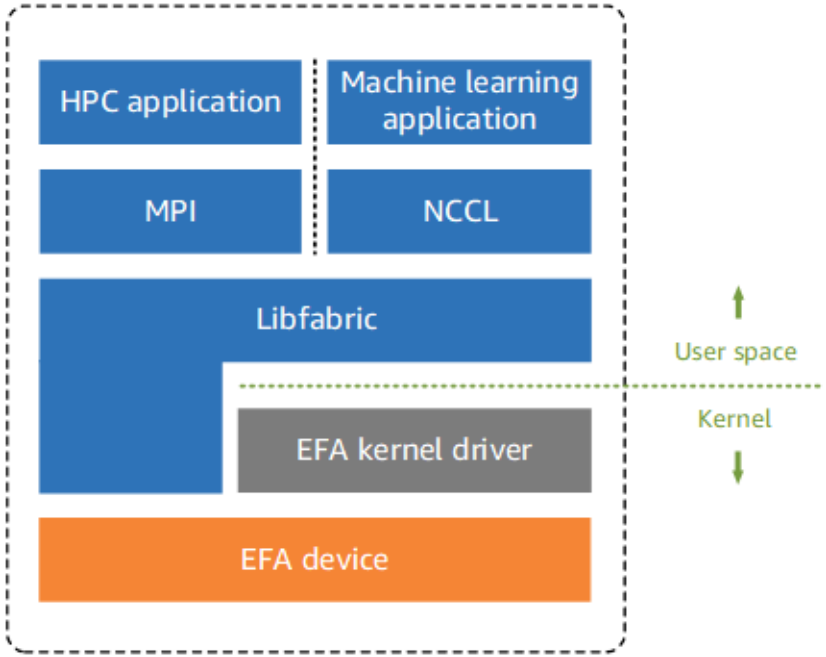
An attempt to approximate an HPC backplane as much as possible in the cloud

Sandia National Laboratories

EFA Layer Cake



Traditional HPC software stack in EC2



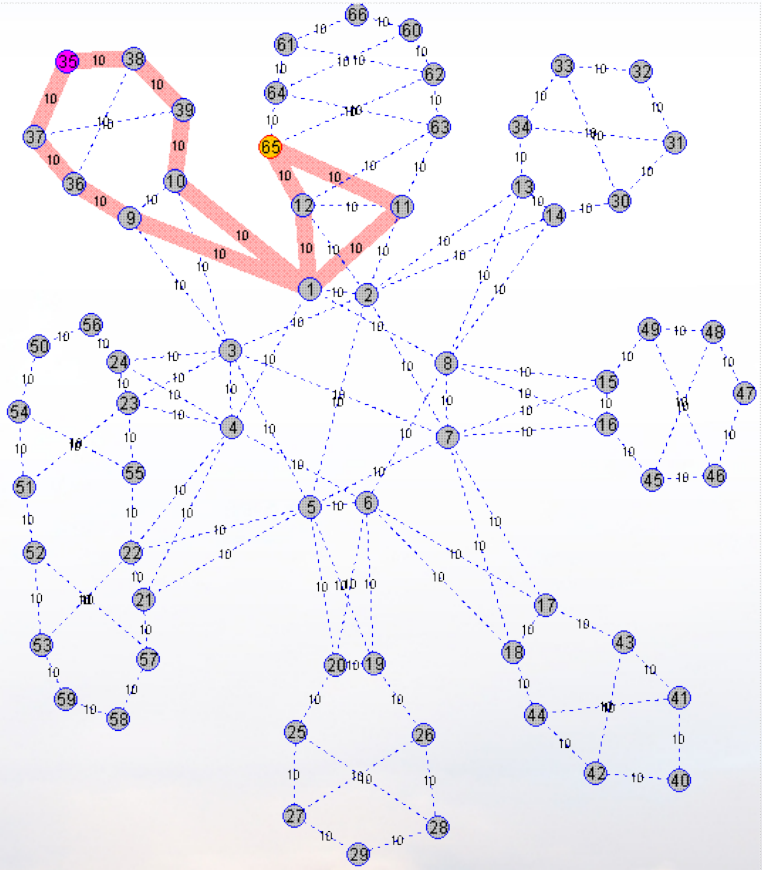
HPC software stack in EC2 with EFA

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa.html

Sandia National Laboratories

SRD Reduces Network Flow Congestion

- **SRD enables Equal-cost Multi-Path routing (ECMP)**
  - Reduces network flow congestion (and thus collisions) by load-balancing traffic over multiple paths
  - Results in much greater network throughput
  - The animated GIF shows equal cost paths between source (purple) and destination (yellow)



https://www.wikiwand.com/en/IEEE_802.1aq

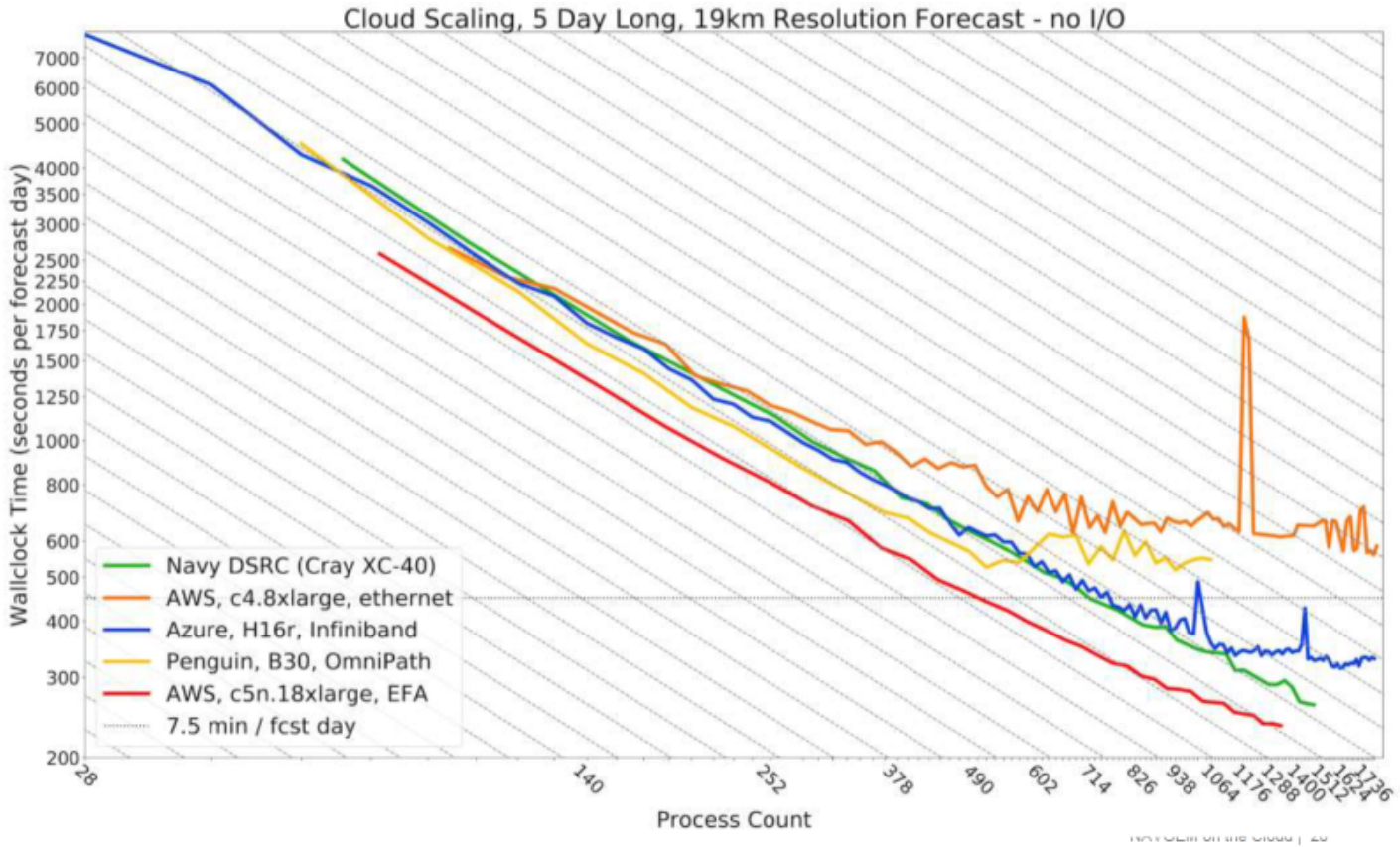
Linear Scaling with EFA

U.S. NAVAL RESEARCH LABORATORY

High Resolution Forecast: Performance - Comparison

Performance Improvements:
C5n with EFA on AWS EC2

- At the highest core counts:
  - 107% faster than Penguin
  - 43% faster than Azure
  - 160% faster than previous AWS
  - 25% faster than Navy DSRC
- Min size estimated to meet 7.5 min:
  - 33% faster than Azure
  - 23% faster than Navy DSRC
- Min size forecast cost estimate:
  - Azure: $82.97
  - C5n with EFA: $44.02



U.S. Naval Research Laboratory

https://www.slideshare.net/insideHPC/navgem-on-the-cloud-computational-evaluation-of-cloud-hpc-with-a-global-atmospheric-model

Sandia National Laboratories

Quantum Computing

Sandia National Laboratories

Quantum Computing Introduction

- **Quantum computing is performed using quantum bits (Qubits), which exist in multiple simultaneous states [0:1]**
  - This phenomenon is called quantum superposition and is fundamentally a parallel processing play
  - Qubits are notoriously unstable and only operative during very short time windows
    - In the quantum world, if you measure it you change it
  - PKI as we know it would no longer be effective in a quantum world (cf. Shor's Algorithm)
- **Quantum entanglement occurs when some groups of simultaneous states correlate or interact with other groups of simultaneous states**

Simultaneous states are not independent; entanglement can operate even at a distance.

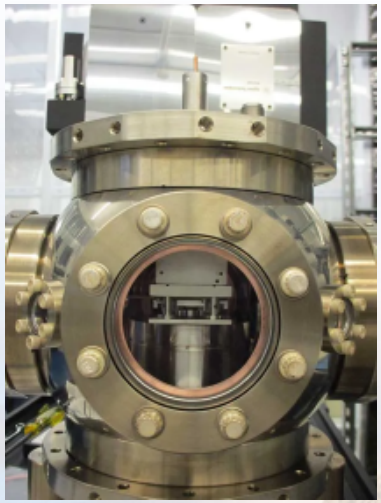
Sandia National Laboratories

Quantum Computing Flows

- **AWS Braket flow (to "explore" quantum)**
  - First, simulate and refine a quantum circuit on conventional hardware using a Jupyter Notebook coded in Python
  - Then schedule a circuit computation notebook cell on quantum hardware, a.k.a. QPU (Quantum Processing Unit)

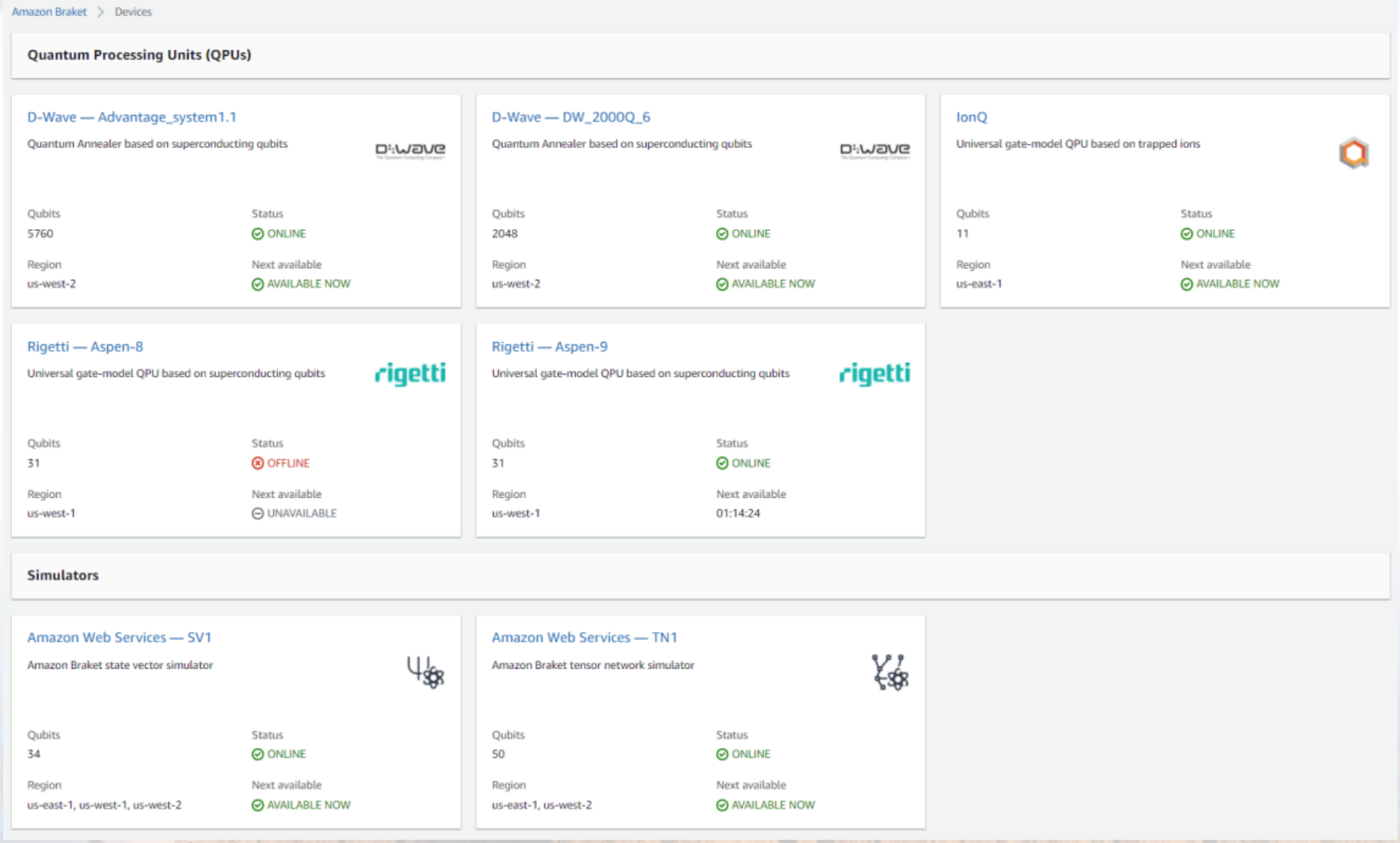
Amazon Braket

- **Sandia Quantum flow, using its own ion-trap-based quantum computing capability recently made externally available, called QSCOUT.**
  - A proposal must be submitted and accepted to access
  - QSCOUT is coded in Jaqal quantum assembly language (Just another quantum assembly language)
  - Install the open source package JaqalPaq with samples, emulators, and transpilers from other quantum languages
  - Python Jupyter Notebooks emit Jaqal code



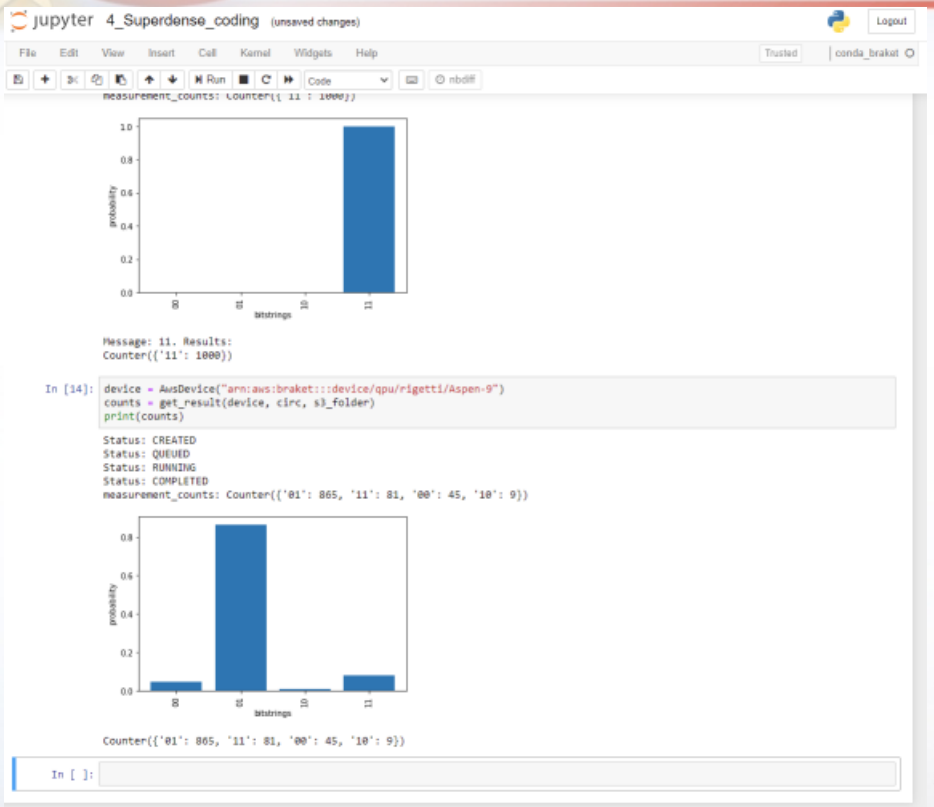
Sandia National Laboratories

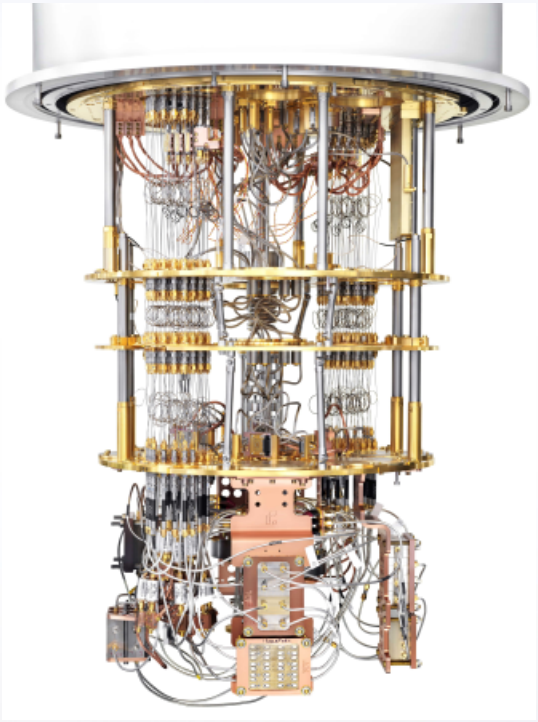
QPUs and Simulators on AWS Braket



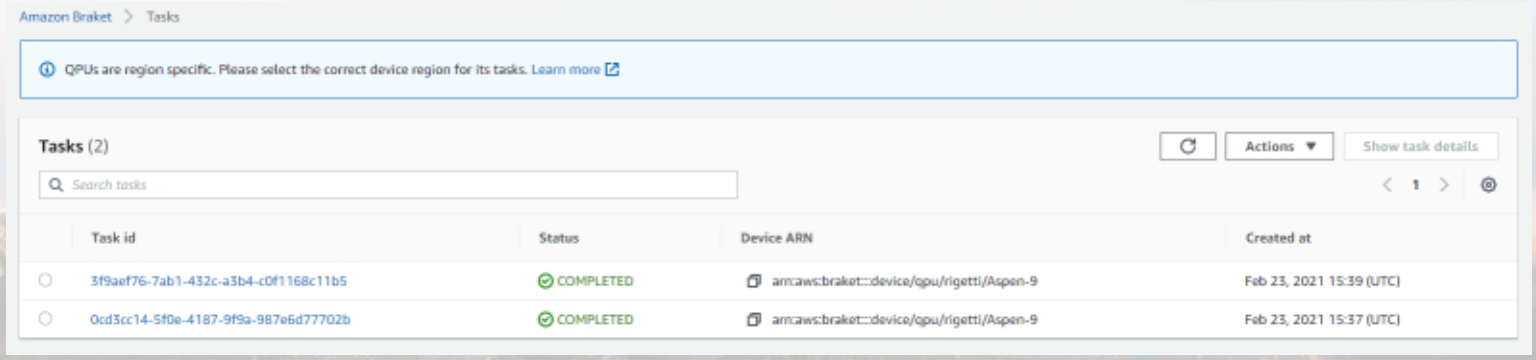
Sandia National Laboratories

After Simulation, Run Cell on QPU





Source: https://aws.amazon.com/braket/hardware-providers/rigetti/



Machine Learning (ML)

Sandia National Laboratories

The 5 Questions that ML Can Answer

Source: Microsoft
Published: March 2019

April 19, 2021

The 5 questions machine learning can answer

Data science predicts answers to questions using algorithms.

It's helpful to think about an algorithm as a recipe and your data as the ingredients. An algorithm tells how to combine and mix the data in order to get an answer. Computers are like a blender. They do most of the hard work of the algorithm for you and they do it pretty fast.

It might surprise you, but there are only five questions that data science answers:

Question	Algorithm family	Example
Is this A or B?	Classification	Will this copier fail in the next two months? Yes or no?
Is this weird?	Anomaly detection	Is this credit card charge normal?
How much - or - How many?	Regression	What will the Q1 expenses be?
How is this organized?	Clustering	Which car models have the most brake problems?
What should I do next?	Reinforcement learning	For a humidity control system: adjust humidity or leave as is?

Types of Learning:

- Supervised (e.g., classification)
- Unsupervised (e.g., clustering)
- Reinforcement (e.g., robotics)

Sandia National Laboratories

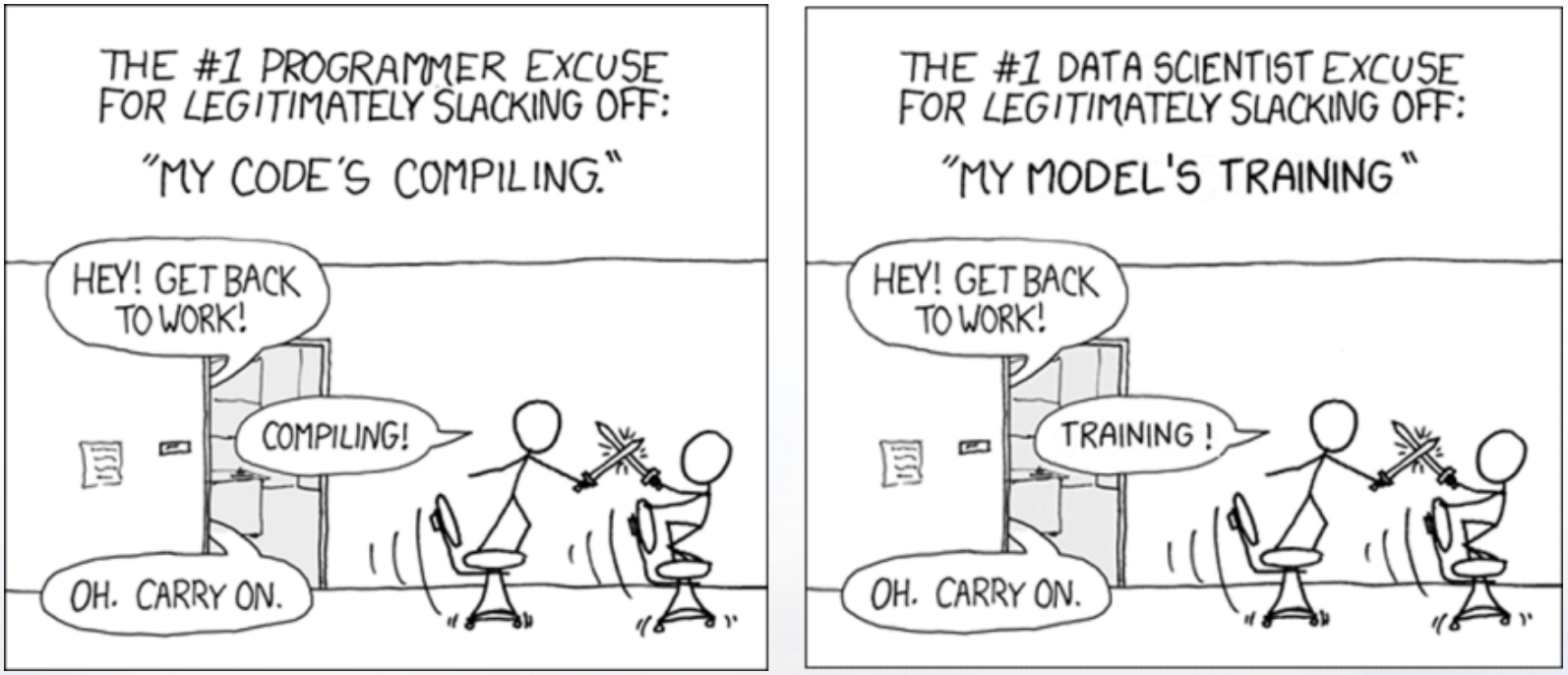
The Data Science Work Flow

It's all about the data. Analyzing the data, visualizing the data, cleaning the data, handling missing or suspect data, transforming the data. Stated another way, data is primary, processing is secondary.

- 0. Prepare data (training, test, holdout) in ML-ready form
- 1. Start with a simple regression or classification model
- 2. 1D viz to inspect features for potential transformation
- 3. Rerun the simple model
- 4. 2D viz to compare features pair-wise for interactions
- 5. Rerun the simple model
- 6. 3D viz to see interactions between features and target
- 7. Rerun the simple model
- 8. Consider engineering (synthesizing) new features
- 9. Rerun the simple model
- 10. Dimensionality reduction and/or feature selection
- 11. Rerun the simple model
- 12. Repeat steps 1-11 using different estimators and deep learning approaches (i.e., transcend the simple model)
- 13. Evaluate final model against the holdout dataset
- 14. Save your model. General rejoicing.

Sandia National Laboratories

Computer Science vs. Data Science



Source: xkcd

ML Technology Landscape

In static graphs, the layers define the computational graph in advance and data is injected at runtime. In dynamic graphs, the computational graph is defined on-the-fly by the forward computation.

- **"Traditional" (generally regression-based) Machine Learning**
  - scikit-learn (the ML Swiss Army Knife)
  - xgboost
- **Deep Learning (neural network-based)**
  - "Deep" refers to the multiple layers in the neural network
  - Front-end APIs (which define model layers)
    - Keras (used in TensorFlow; now deprecated in MXNet)
    - Gluon (used in newer releases of MXNet)
  - Back-end Frameworks (which run data through layers)
    - PyTorch (Facebook; dynamic graph; more used for research)
    - TensorFlow (Google; static graph; more used for production)
    - MXNet (Apache; popular with AWS, which generally uses Keras)
- **The use of Jupyter Notebooks in Python is pervasive in data science**

Sandia National Laboratories

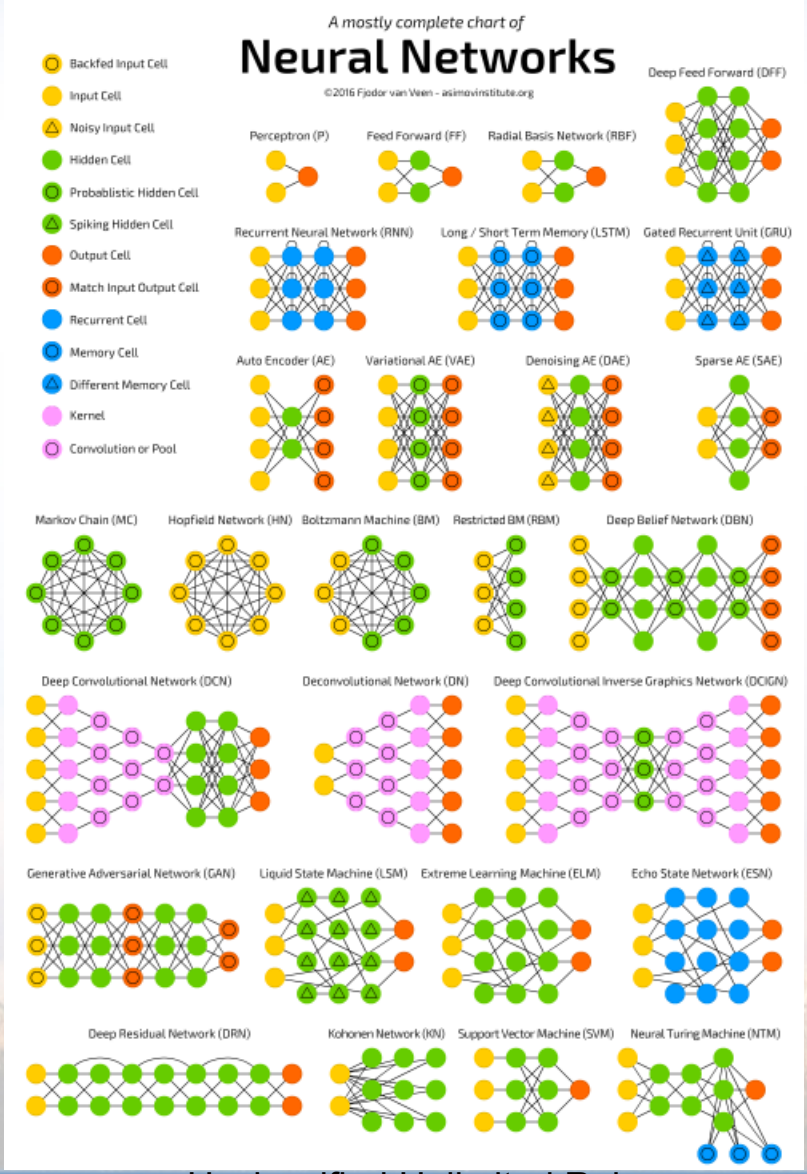
Popular Types of Neural Networks

Neural networks are biologically inspired (neurons and synapses)

- **Multi-Layer Perceptron (MLP)**
  - Fully-connected, multi-layer neural network good for speech recognition and machine translation
- **Convolutional Neural Network (CNN)**
  - 3D arrangement of neurons good for classifying images
- **Recurrent Neural Network (RNN)**
  - Preserves temporal sequence information; good for text processing, speech recognition, and handwriting recognition.
- **Long Short-Term Memory (LSTM)**
  - RNN variant preserves long-term memory; good for speech.
- **Generative Adversarial Network (GAN)**
  - Generates “deep fake” exemplars from CNN models
- **Deep Learning CNNs were my ML focus**

Sandia National Laboratories

Neural Network Chart



https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

Installing Deep Learning Frameworks

- **Forward and backward passes through deep network layers are implemented using matrix-vector multiplication**
- **This is fundamentally parallel, so GPUs are commonly used.**
- **Unfortunately, each leading Deep Learning Framework (PyTorch, TensorFlow, MXNet) requires a different version of NVIDIA's CUDA and cuDNN**
  - Makes it difficult (but not technically impossible) to support multiple frameworks on the same machine
- **AWS takes the curse out of that "undifferentiated heavy lifting"**

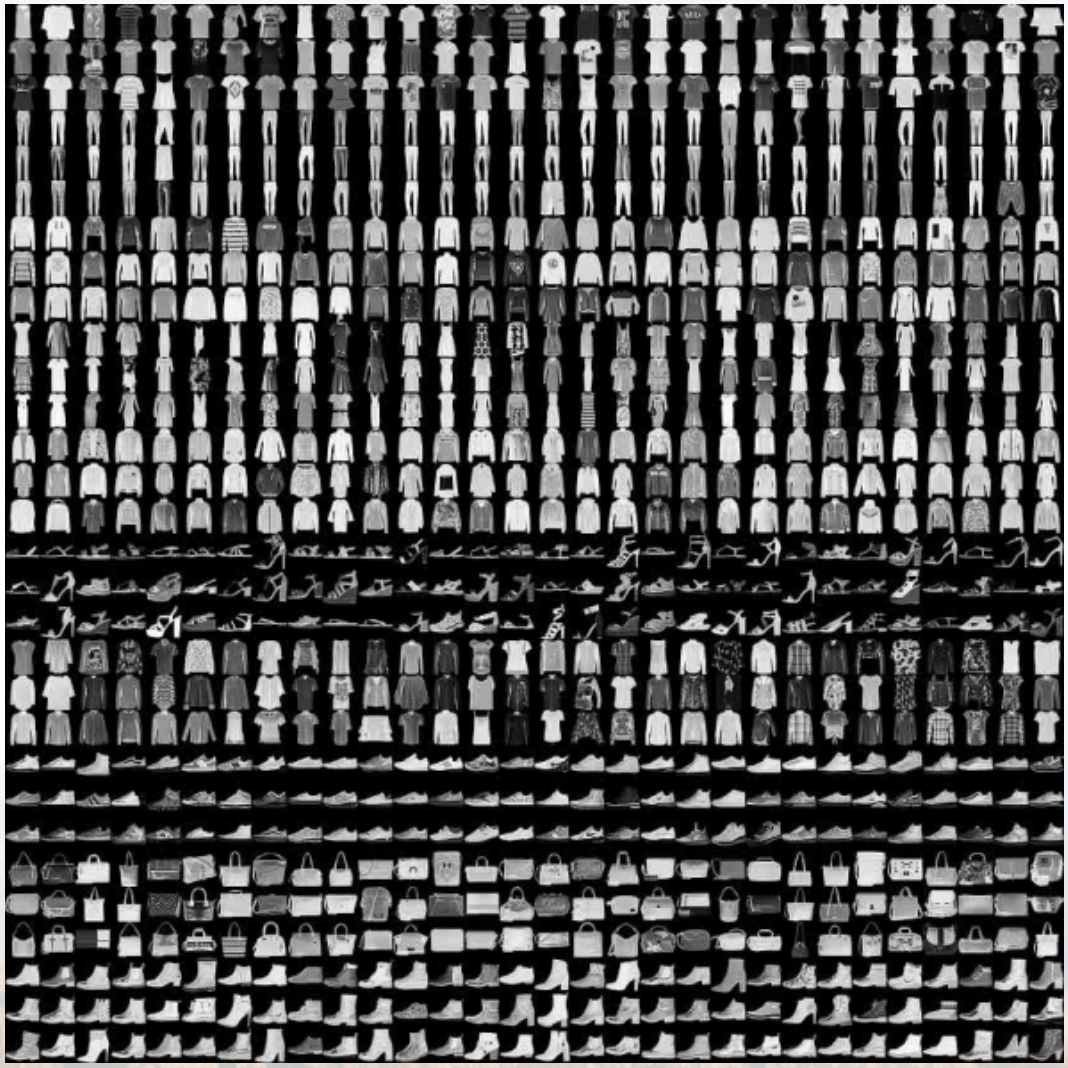
Options for Deep Learning on AWS

- **Launch a GPU-enabled EC2 instance using a Deep Learning AMI for the desired OS from AWS Marketplace**
  - SSH to the instance and invoke Jupyter Notebook. Choose kernel with desired framework and language version.
  - Some environments (i.e., classified) may need to tunnel UI through bastion host and require security group changes.
- **Launch a GPU-enabled Jupyter Notebook instance from AWS SageMaker**
  - The only option that worked for me on the classified cloud
  - Requires special tag values and IAM Access Role
- **Load the desired framework and language kernel from SageMaker Studio**

SageMaker Studio is the most convenient and serverless, but currently only supported on AWS Commercial

Sandia National Laboratories

MNIST/Fashion-MNIST to Learn CNNs



https://deeplizard.com/learn/video/EqpzfvxBx30

Sandia National Laboratories

Several ML CNN Prototypes Created

- **MNIST CNN on TensorFlow using Keras in SageMaker Studio Jupyter Notebook for *model***
  - Two different layer structures used for the Keras CNN model; the first attempt overfit but the second didn't.
- **MNIST CNN on TensorFlow using Keras in SageMaker Notebook instance for *tuning***
  - Required because of a SageMaker Studio container issue
  - Completely restructured the Jupyter Notebook code to create endpoint and perform hyperparameter tuning
    - Note that hyperparameter tuning (which takes about a half hour to run) does *not* tune or change the layers of the model
  - Real-time inference with hand-drawn digits using endpoint
- **MNIST CNN on PyTorch in SageMaker Studio for *comparison***
  - Observed the differences in the way that dynamic model graphs and static model graphs are coded

Hyper-parameter tuning just iterates over hyper-parameter ranges and scrapes output results

Sandia National Laboratories

SageMaker Notebook For Tuning

jupyter keras-mnist-sagemaker-hyperparameter-endpoint Last Checkpoint: 02/01/2022 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted conda_tensorflow_p36

Run Markdown nbdiff

Find the best hyperparameters with Automatic Model Tuning

This is by far the most expensive part of this exercise; we're going to use a fair amount of time on P3 instances here. If you're worried about your AWS costs, skip the rest of this notebook and just shut down your SageMaker notebook instance now. Tuning took about a half hour.

In [13]:
```
from sagemaker.tensorflow import TensorFlow

tf_estimator = TensorFlow(entry_point='mnist-train-cnn.py',
                          role=role,
                          instance_count=1,
                          instance_type='ml.p3.2xlarge',
                          framework_version='2.2',
                          py_version='py37',
                          script_mode=True
                         )
```

In [14]:
```
from sagemaker.tuner import IntegerParameter, CategoricalParameter, ContinuousParameter, HyperparameterTuner

hyperparameter_ranges = {
    'epochs':        IntegerParameter(5, 20),
    'learning-rate': ContinuousParameter(0.0001, 0.1, scaling_type='Logarithmic'),
    'batch-size':    IntegerParameter(32, 1024),
}

objective_metric_name = 'val_acc'
objective_type = 'Maximize'
metric_definitions = [{'Name': 'val_acc', 'Regex': 'val_accuracy: ([0-9\\.]+)'}]

tuner = HyperparameterTuner(estimator=tf_estimator,
                            objective_metric_name=objective_metric_name,
                            metric_definitions=metric_definitions,
                            hyperparameter_ranges=hyperparameter_ranges,
                            max_jobs=10,
                            max_parallel_jobs=2,
                            objective_type=objective_type)
```

In [ ]:
```
tuner.fit({'training': training_input_path, 'validation': validation_input_path})
```

Deploy the best model

In [ ]:
```
# Look at the Hyperparameter tuning jobs menu under Training on the SageMaker Dashboard to find out what was measured "best".
# Namely, the model with the max val_accuracy picked by the tuner and the set of hyperparameter values that made it the best.
import time

tf_endpoint_name = 'keras-tf-mnist-' + time.strftime("%Y-%m-%d-%H-%M-%S", time.gmtime())

# Note that TensorFlow version 2.2 doesn't support EIA, so elastic inference accelerators cannot be used.
# Also note that this command always generates the warning below, which seems benign:
# update_endpoint is a no-op in sagemaker>=2.
# See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
tf_predictor = tuner.deploy(initial_instance_count=1,
                            instance_type='ml.c5.large',
                            endpoint_name=tf_endpoint_name)
```


Sandia National Laboratories

SageMaker Autopilot (AutoML)

- **SageMaker Autopilot automatically selects the combination of data preprocessing and machine learning algorithm that best fits your dataset**
- **Basically, AutoML applies Machine Learning to Machine Learning**
- **Constraints**
  - Regression or classification problems; not for deep learning.
  - Data available in CSV format (with header row) in S3
- **SageMaker Experiments used to iterate over potential preprocessing & algorithms**
- **Can deploy the best model as an endpoint for real-time inference**

F1 was the metric (a weighted average of Precision and Recall)

Took about an hour and a half using the default of 250 models

Sandia National Laboratories

SageMaker Autopilot Prototype

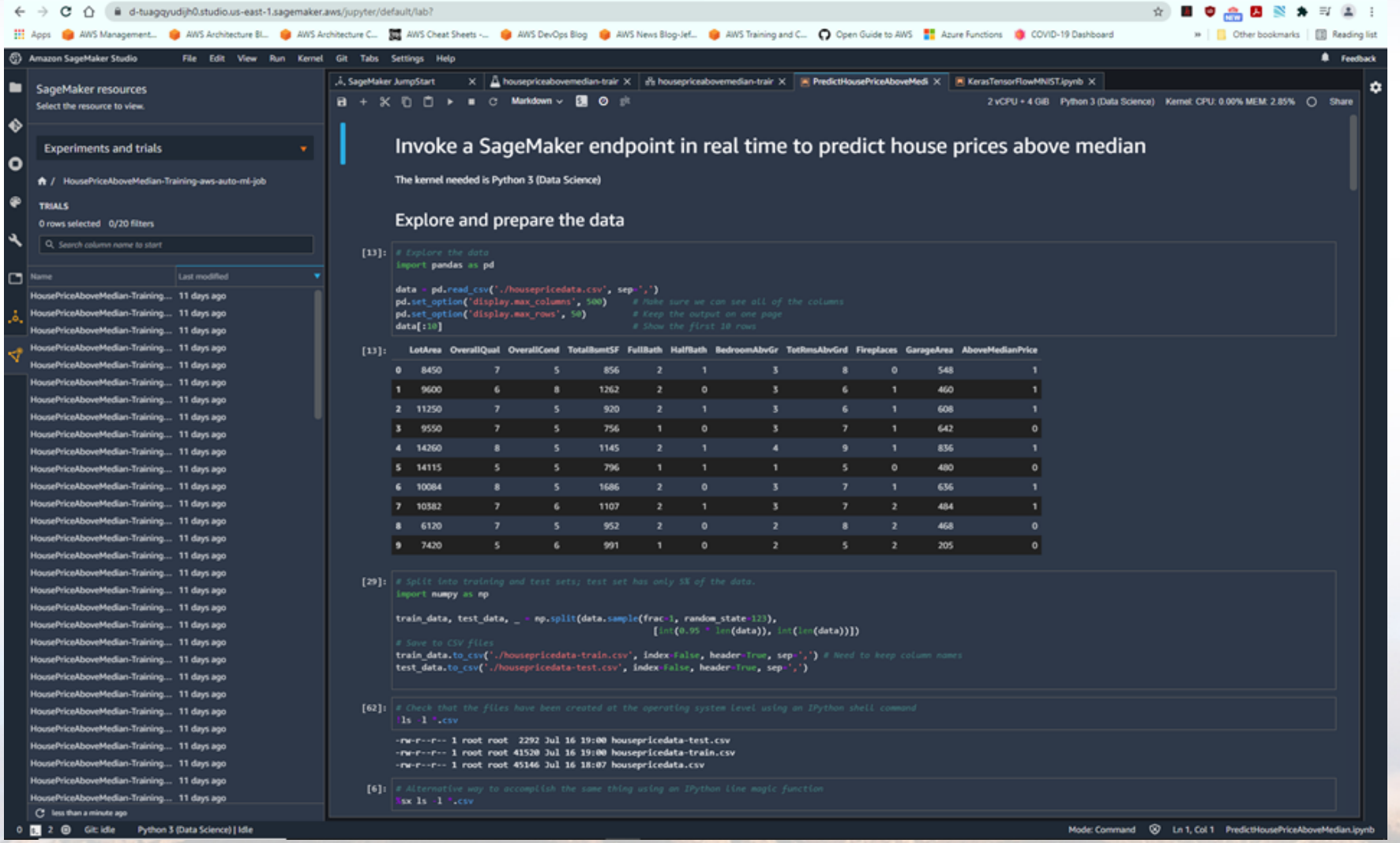
- Created a SageMaker Autopilot (AutoML) experiment using a public domain housing price CSV data set from Kaggle
- Deployed an endpoint based on the best model out of 250 iterations
- Performed real-time inference on a test set using that endpoint
- Calculated the Confusion Matrix

```
Confusion Matrix
38 5
1 29

Accuracy=0.9178, Precision=0.8529, Recall=1.2667, F1=1.0194
```

Sandia National Laboratories

SageMaker Autopilot in Studio



	LotArea	OverallQual	OverallCond	TotalBsmtSF	FullBath	HalfBath	BedroomAbvGr	TotRmsAbvGrd	Fireplaces	GarageArea	AboveMedianPrice
0	8450	7	5	856	2	1	3	8	0	548	1
1	9600	6	8	1262	2	0	3	6	1	460	1
2	11250	7	5	920	2	1	3	6	1	608	1
3	9550	7	5	756	1	0	3	7	1	642	0
4	14260	8	5	1145	2	1	4	9	1	836	1
5	14115	5	5	796	1	1	1	5	0	480	0
6	10084	8	5	1686	2	0	3	7	1	636	1
7	10382	7	6	1107	2	1	3	7	2	484	1
8	6120	7	5	952	2	0	2	8	2	468	0
9	7420	5	6	991	1	0	2	5	2	205	0

Takeaways

Takeaways

- AWS is representative of most cloud vendors and provides leading-edge capabilities in the cloud. However, only the ML capabilities—and only a subset—are currently available in both GovCloud and classified clouds.
- These capabilities are most useful when an organization already has a significant amount of data in the cloud
- However, the nature of the cloud is that it provides a flexible and agile sandbox for experimentation, exploration, and prototyping
- IoT computing is applicable when large numbers of sensors need to be processed and aggregated
- Edge computing capabilities are applicable to factory floor or fabrication environments, as well as in-theater battlefield contexts and sensor-based workgroups in general
- HPC and Quantum Computing capabilities may be less useful at National Laboratories that have internal access to such computing resources
- However, the rich set of Machine Learning capabilities in the cloud may prove very useful for rapid prototyping and domain exploration, and potentially to expose endpoints for real-time inference

ML Demo

Sandia National Laboratories

Q&A

Backup

Sandia National Laboratories