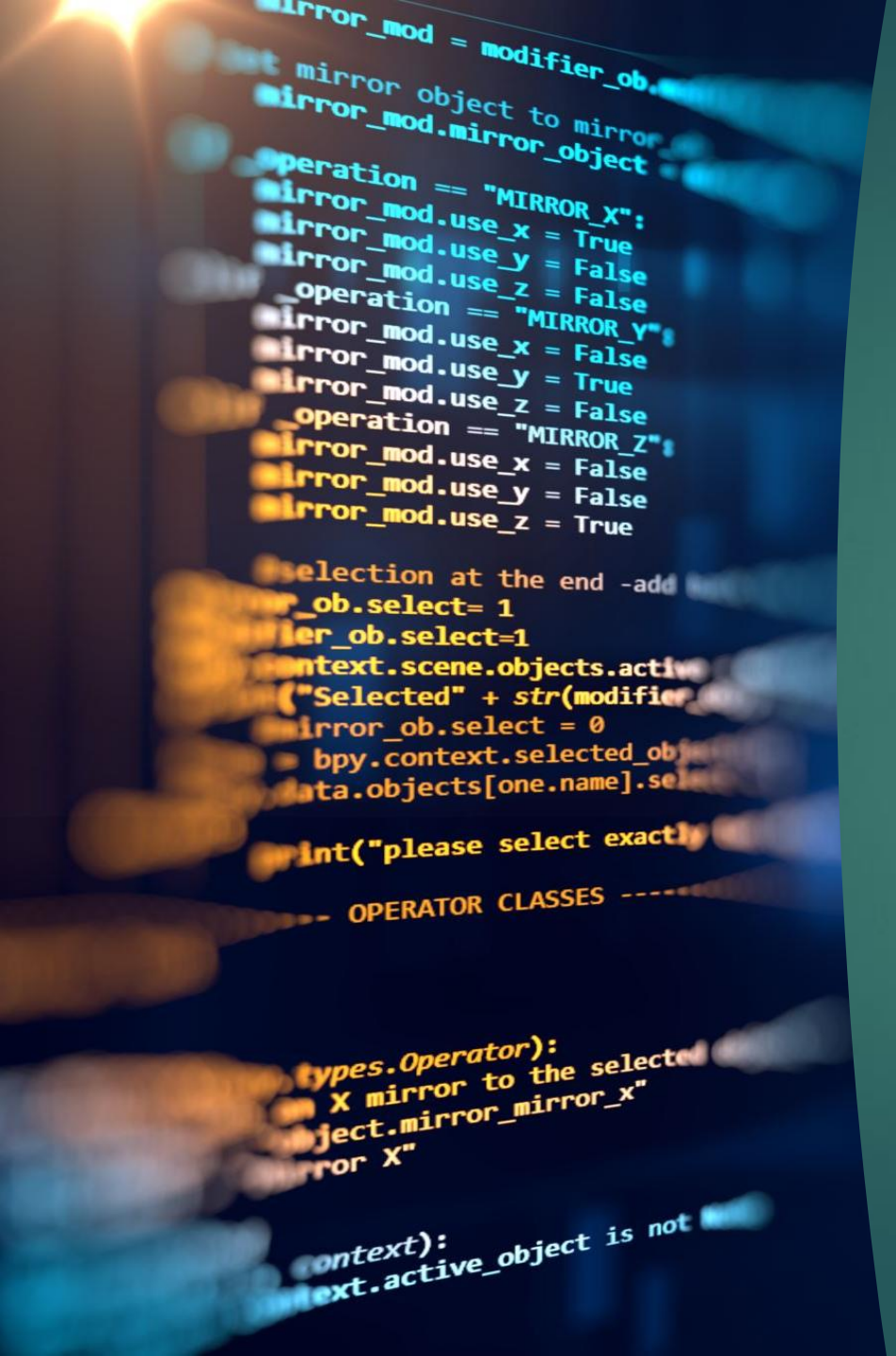


Processing in the Cloud

NETOPS XI SESSION 2, NOVEMBER 2, 2021

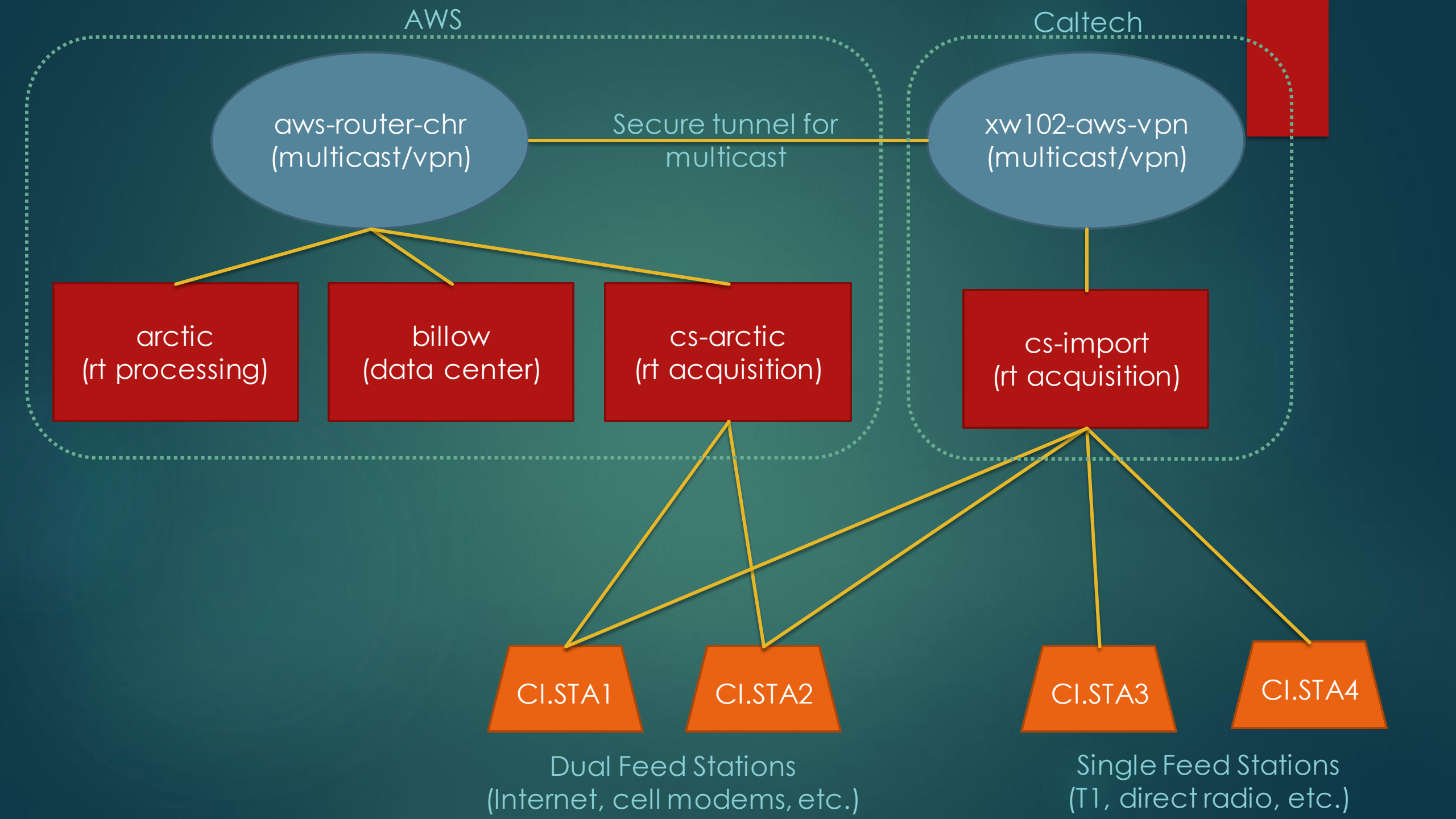
MODERATORS: SHANG-LIN CHEN AND ELLEN YU

Follow the Data

- Real-time acquisition
- Data processing
- Data and product storage
- Data and product distribution
- Building infrastructure in the cloud (infrastructure as code)

Can we get data from the field to the cloud?

- SCSN
  - VPN shares multicast data with cloud
  - Quakes2AWS
    - Kinesis stream of waveform data


AWS
Caltech
aws-router-chr
(multicast/vpn)
Secure tunnel for
multicast
xw102-aws-vpn
(multicast/vpn)
arctic
(rt processing)
billow
(data center)
cs-arctic
(rt acquisition)
cs-import
(rt acquisition)
CI.STA1
CI.STA2
CI.STA3
CI.STA4
Dual Feed Stations
(Internet, cell modems, etc.)
Single Feed Stations
(T1, direct radio, etc.)

Quakes2AWS pipeline

- **Quakes2AWS** is an AWS-based pipeline that processes continuous, real-time seismic data emitted from stations across Southern California.
  - It uses a machine-learning picker to find picks within the data.

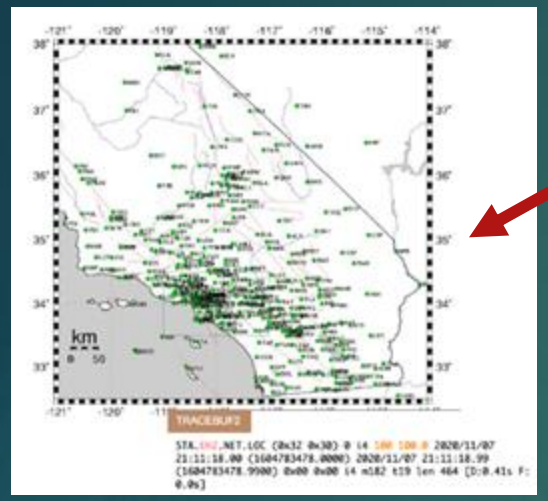
Key Amazon Web Services (AWS) Technologies Used

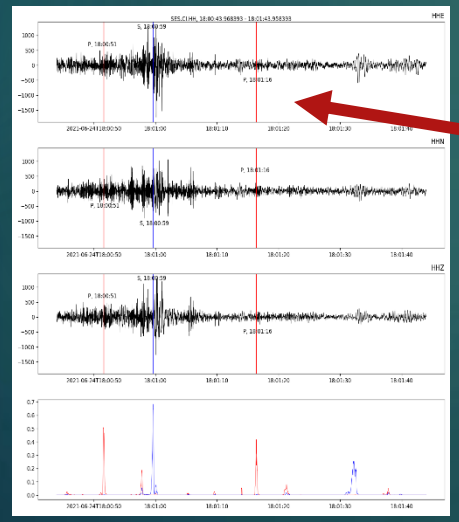
AWS Kinesis Data Stream (KDS)
Amazon Lambda Function
Amazon Relational Database Service (RDS) for PostgreSQL
Simple Storage Service (S3) – Amazon disk storage
Amazon Elastic Container Services (ECS) – container orchestration service

Some Non-AWS Specific Technologies Used

Python 3.6 – high-level programming language
Django 3.0.6 - python based full stack web development framework
PyTorch - open source machine learning library for Python
Docker - designed to make it easier to create, deploy, and run applications by using containers
Terraform - "Infrastructure as code" used for building, changing, and versioning infrastructure safely and efficiently

Quakes2AWS pipeline





quakes2aws-0.5.2 architecture

Station
EH*
HH*

cell
dedicated line
satellite
cable modem
etc.

multicast network

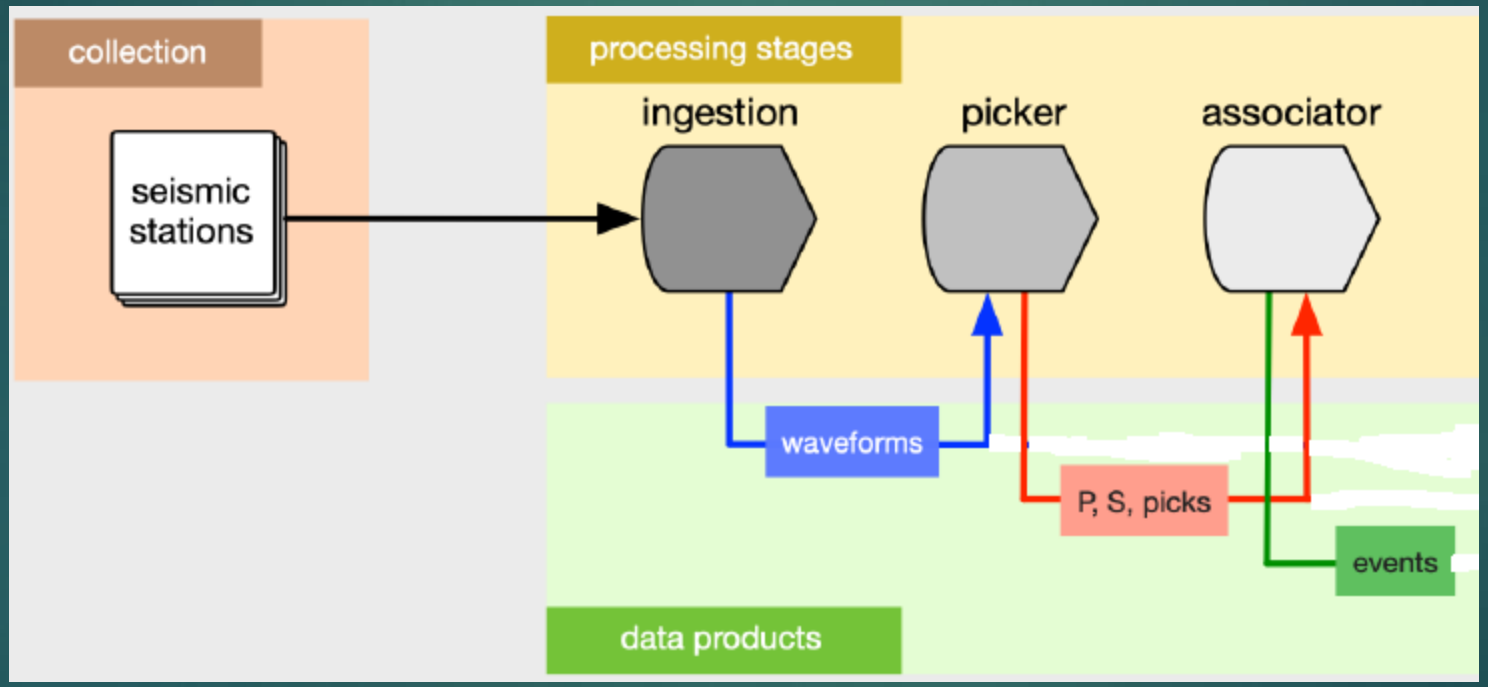
mcast2ew
qmcast2ew
WAVE_RING
ew2aws
aws2ew
PICK_RING
cinnabar

Caltech/USGS

ingest dashboard
S3
incoming data archive
kinesis input
lambda
RDS
postgres 11
seismic data
pick waveforms
grafana dashboard
picker control web application
picker container
pytorch
container machine
picks
kinesis output
ECS cluster

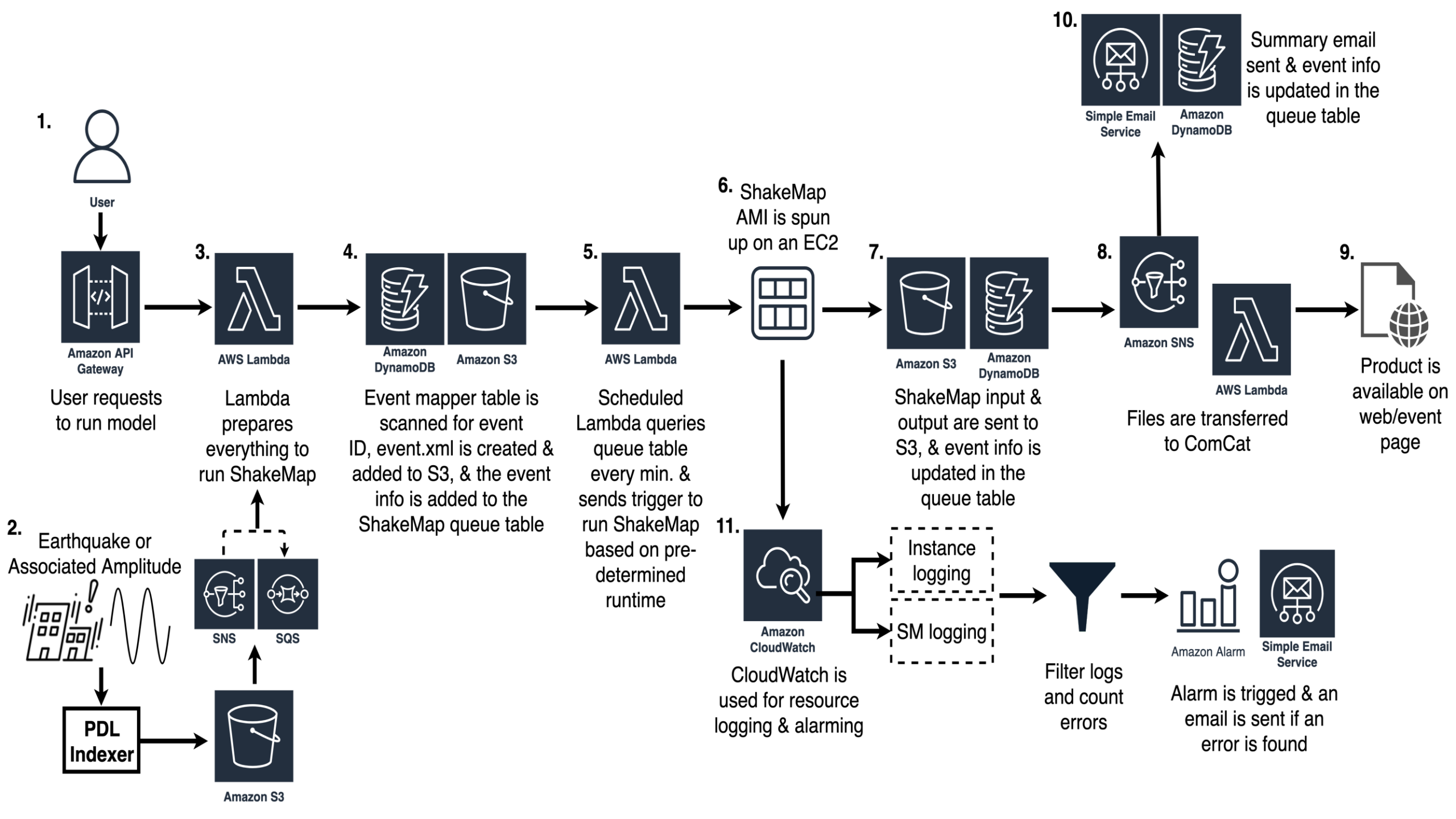
AWS: us-west-2 (oregon)

Quakes2AWS pipeline, at a high level



Can we generate products in the cloud?

- EHP
  - ShakeMap
  - Ground Failure
  - MOTUS
  - Finite Fault
- SCSN
  - AQMS forklifted to the cloud
  - Quakes2AWS lambda functions and picker
- CSN
  - S3 and terrestrial server


1.
User
Amazon API Gateway
User requests to run model
2.
Earthquake or Associated Amplitude
PDL Indexer
Amazon S3
3.
AWS Lambda
Lambda prepares everything to run ShakeMap
SNS
SQS
4.
Amazon DynamoDB
Amazon S3
Event mapper table is scanned for event ID, event.xml is created & added to S3, & the event info is added to the ShakeMap queue table
5.
AWS Lambda
Scheduled Lambda queries queue table every min. & sends trigger to run ShakeMap based on pre-determined runtime
6.
ShakeMap AMI is spun up on an EC2
7.
Amazon S3
Amazon DynamoDB
ShakeMap input & output are sent to S3, & event info is updated in the queue table
8.
Amazon SNS
AWS Lambda
Files are transferred to ComCat
9.
Product is available on web/event page
10.
Simple Email Service
Amazon DynamoDB
Summary email sent & event info is updated in the queue table
11.
Amazon CloudWatch
Instance logging
SM logging
CloudWatch is used for resource logging & alarming
Filter logs and count errors
Amazon Alarm
Simple Email Service
Alarm is trigged & an email is sent if an error is found

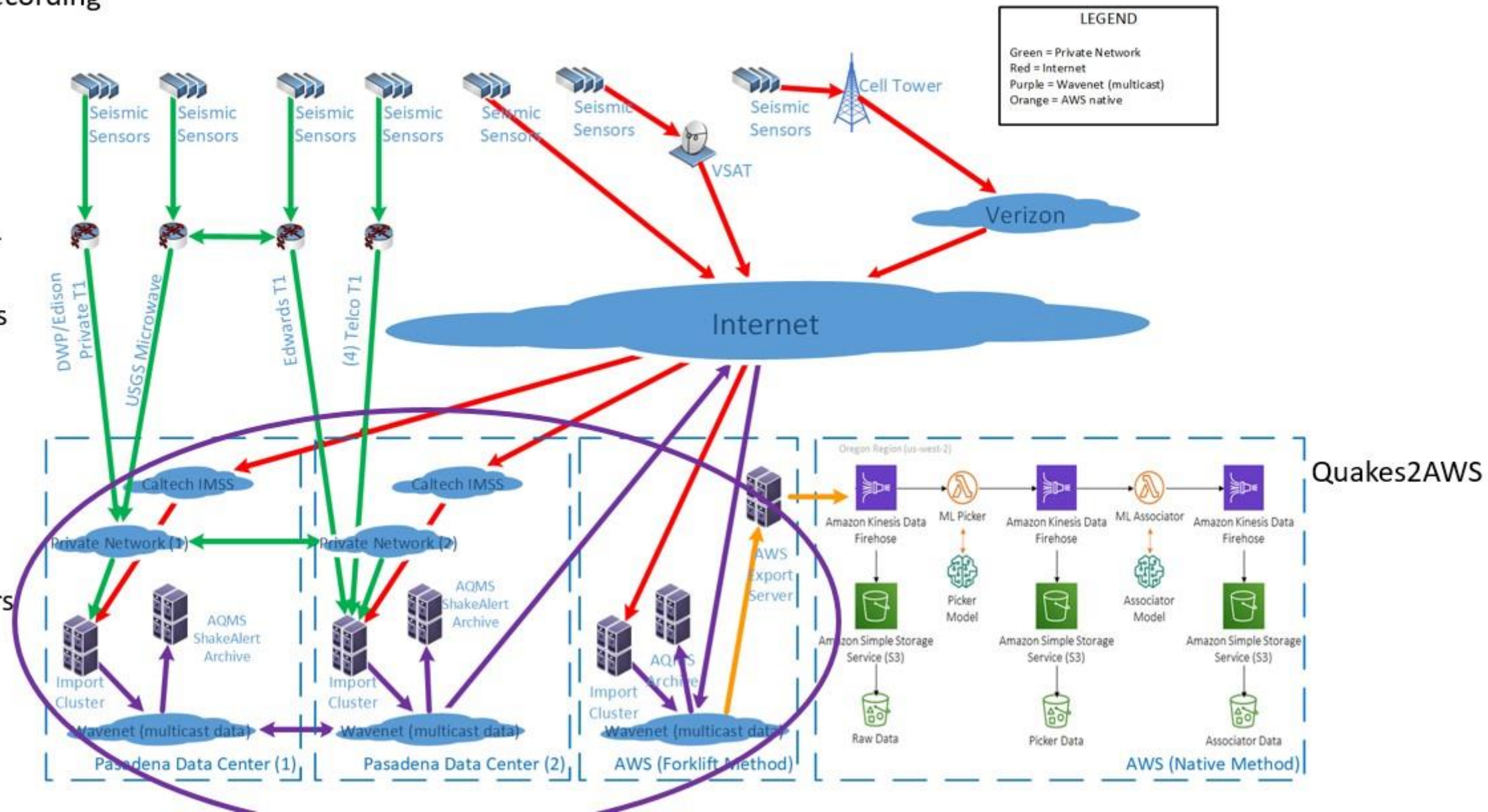
SCSN High-Level Data Flow

Field Data Recording

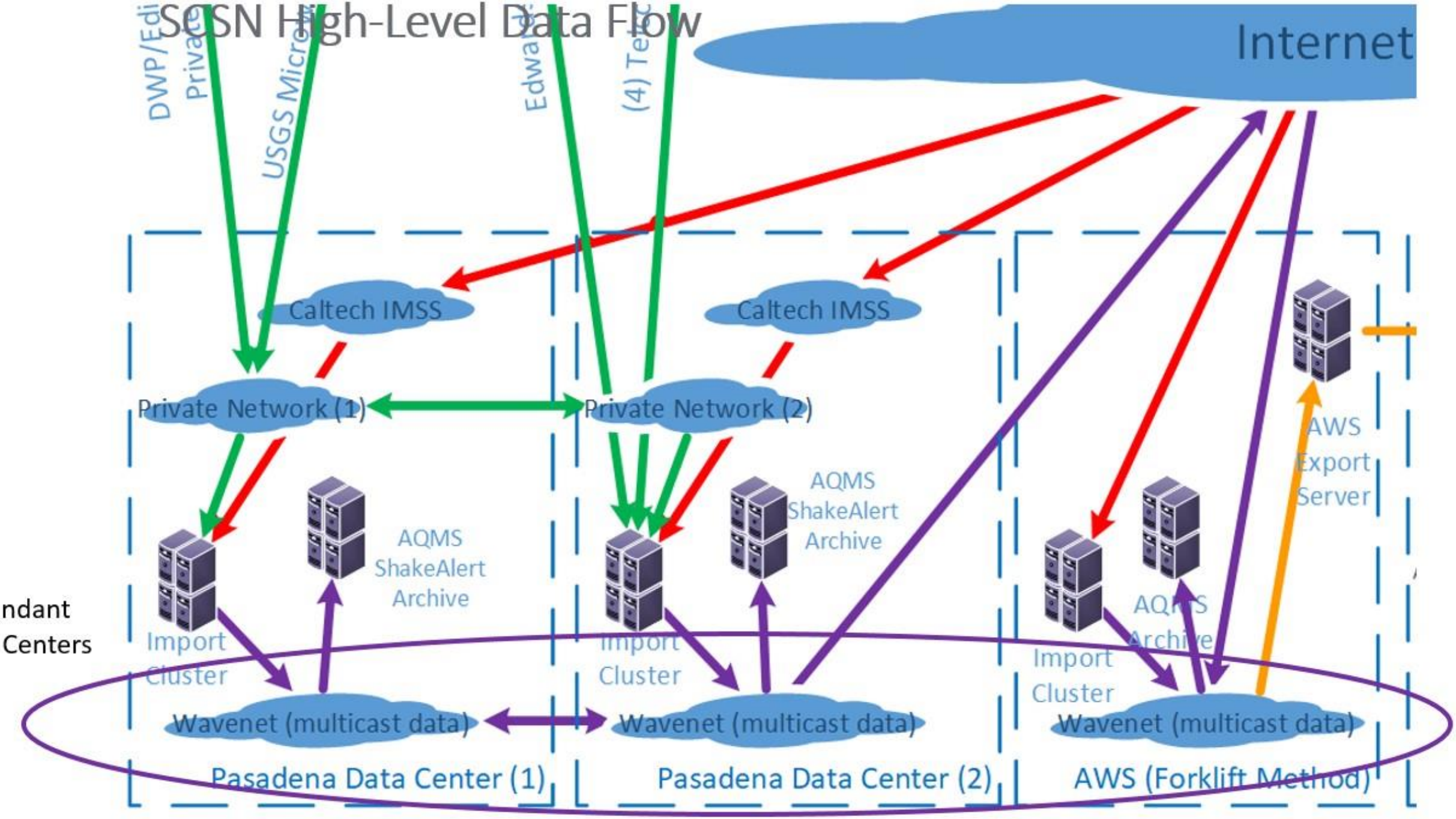
Telemetry – SCSN and Cooperators

Redundant Data Centers

Quakes2AWS



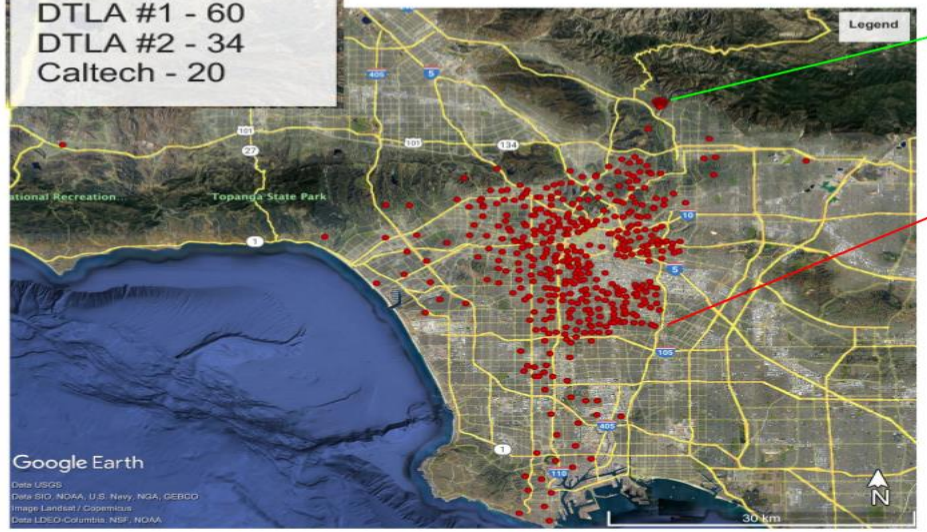
SCSN High-Level Data Flow

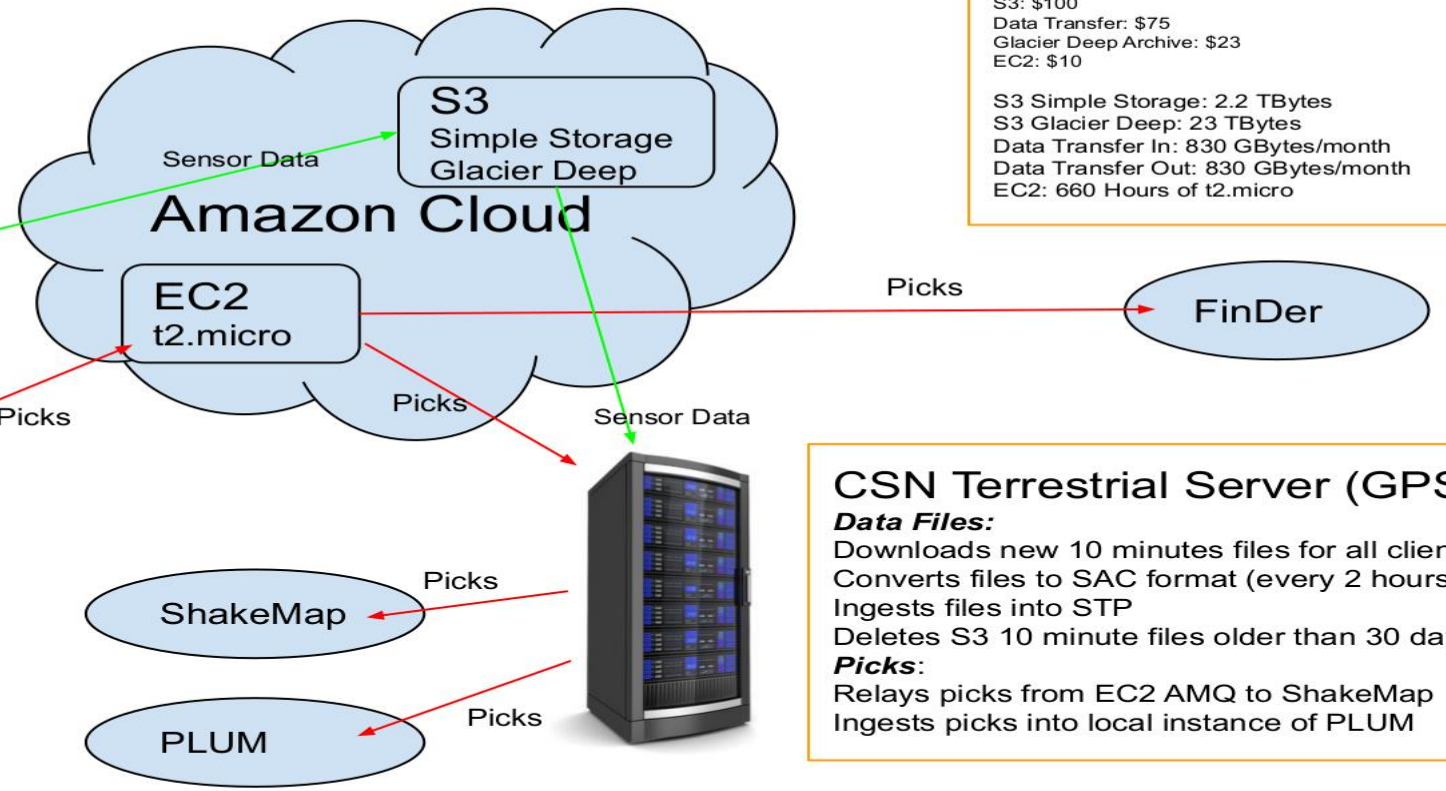


Caltech Community Seismic Network (CSN) Cloud Operations 2021

CSN 2021

LAUSD - 400
JPL - 200
DTLA #1 - 60
DTLA #2 - 34
Caltech - 20





Costs per month (approx.)

S3: $100
Data Transfer: $75
Glacier Deep Archive: $23
EC2: $10

S3 Simple Storage: 2.2 TBytes
S3 Glacier Deep: 23 TBytes
Data Transfer In: 830 GBytes/month
Data Transfer Out: 830 GBytes/month
EC2: 660 Hours of t2.micro

CSN Terrestrial Server (GPS)

Data Files:
Downloads new 10 minutes files for all clients
Converts files to SAC format (every 2 hours)
Ingests files into STP
Deletes S3 10 minute files older than 30 days
Picks:
Relays picks from EC2 AMQ to ShakeMap
Ingests picks into local instance of PLUM

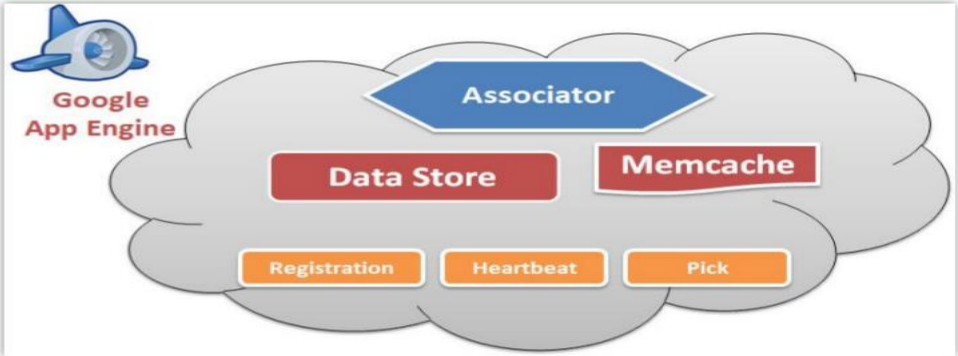
Client Hardware:

Phidgets Tri-axial MEMS accelerometer (16bit), **RPi** 3+, SD card storage, Battery Backup, PSU

Client Software:

Python with ***boto, stompy, gzip*** libraries.
Creates 10 minute text files at 50 sps of NTP corrected timestamped tri-axial acceleration measurements.
Completed files are ***gzip***ped (~300 kBytes) and sent to S3 using ***boto***. (Each client produces **~40 MBytes** of data per day, **1.3 GBytes** per month)
Picker code detects PGA > ½% g and sends AMQ message to EC2 using ***stomp***.

CSN 2011-2017 Google App Engine Architecture



Can we store data and products in the cloud?

- EHP
  - ComCat
- SCSN
  - EC2 database
  - Waveform backups in the cloud (Glacier, Deep Archive)
  - Open Dataset (S3)
- CSN
  - S3, Glacier

Waveform Backups

- Cloud backups of waveforms in addition to offsite physical backups
- Started with Glacier, switched to Deep Archive because of cost and ease of use
- Advantages:
  - No physical media
  - Offsite but no shipping needed
- Disadvantages:
  - Need to move some existing backups to Deep Archive because DA is cheaper
  - Costs grow
  - Cost of recovery

Open Dataset

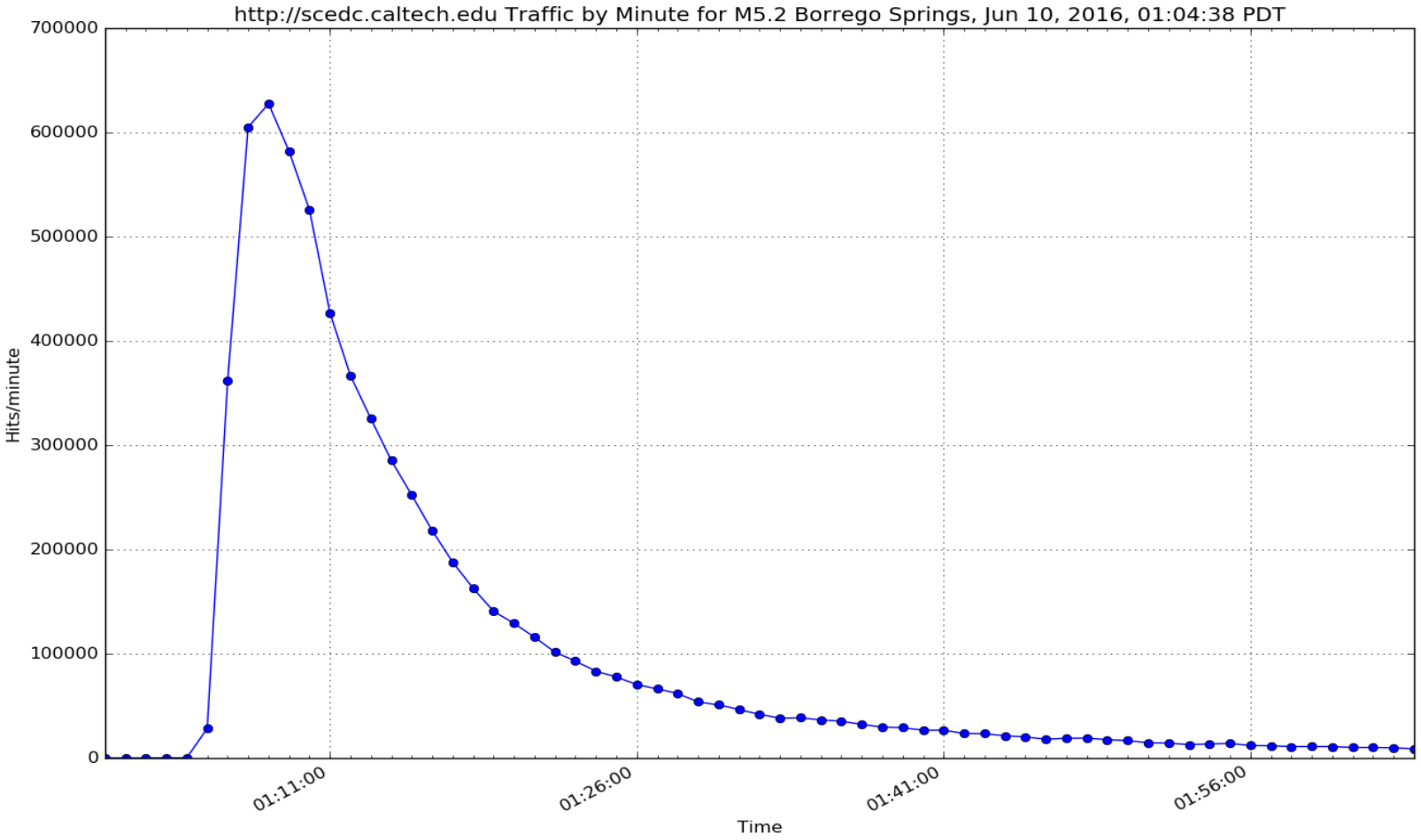
- Amazon sponsors publicly available datasets in the cloud
  - Amazon pays for S3 storage and transfer
  - Users can download data or ingest into their own cloud applications
- Renewable every 2 years
- SCSN continuous and triggered waveform data, event catalogs, and phase picks are available in an Open Dataset
  - Already renewed once

Can we distribute data products from the cloud?

- EHP
  - Real-time feeds
  - Event pages, map and list, and hazard tools
- SCSN
  - Website (scedc.caltech.edu, scsn.org)
  - Open Dataset lambda functions
- PNSN
  - Website (pnsn.org)
  - SQUAC

SCEDC Website in the Cloud

- Pre-cloud: Apache web servers hosted on two physical servers
  - Unresponsive when large events occurred
- Moved static portion of website to S3
  - Statically generated earthquake maps that were the main cause of traffic spikes
  - Other HTML, CSS, Javascript, images
- Added Cloudfront
  - Allowed custom TLS certificate
  - Lower data transfer costs
  - Geographic caching close to users for infrequently updated pages
  - ~$17/month

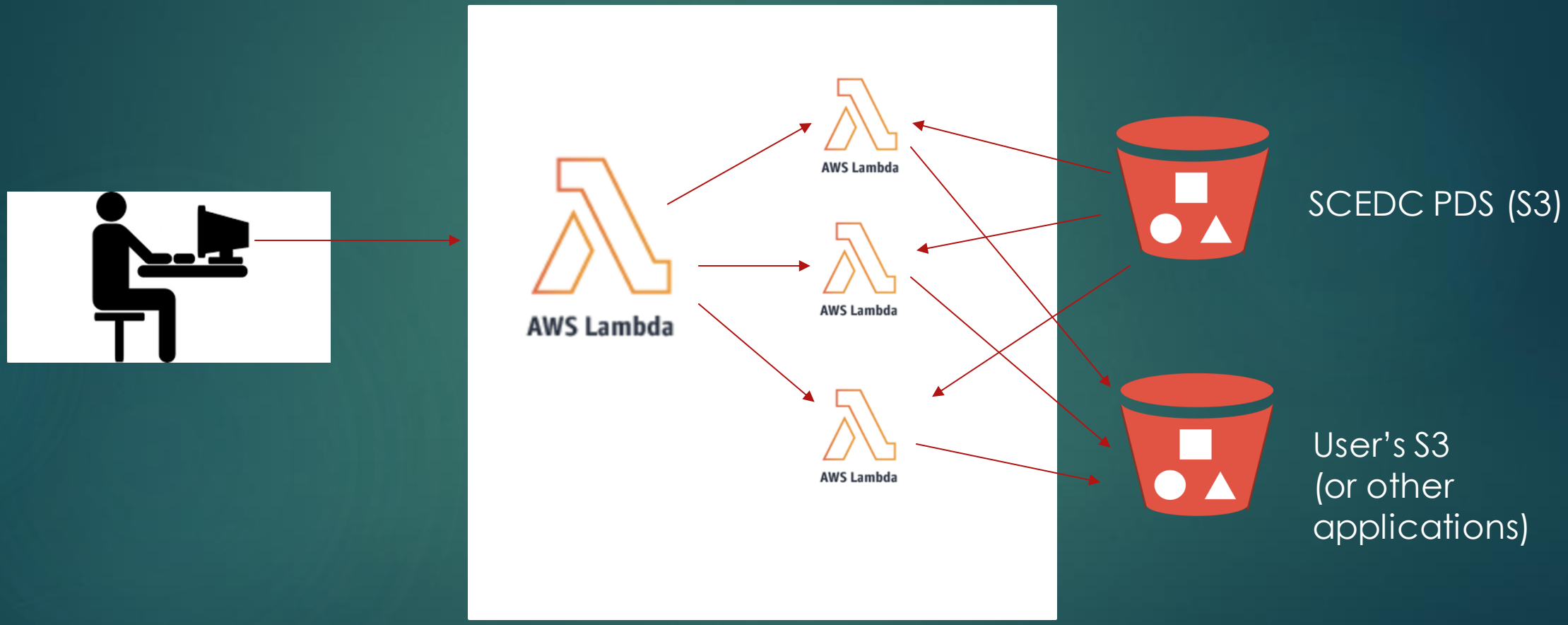

http://scedc.caltech.edu Traffic by Minute for M5.2 Borrego Springs, Jun 10, 2016, 01:04:38 PDT
700000
600000
500000
400000
300000
200000
100000
0
Hits/minute
01:11:00
01:26:00
01:41:00
01:56:00
Time

Distributing Data from the Open Dataset

- Download to EC2 or non-cloud computer using tools (awscli, s3fs) and libraries (boto3)

- Users can write their own serverless functions (Lambda)
  - Process data before using it
  - Write data to their own S3 bucket or feed to another function or container
  - Create API with API Gateway
  - Use triggers
  - Don't need to provision servers

User-Defined Lambda Functions



PNSN Cloud Architecture

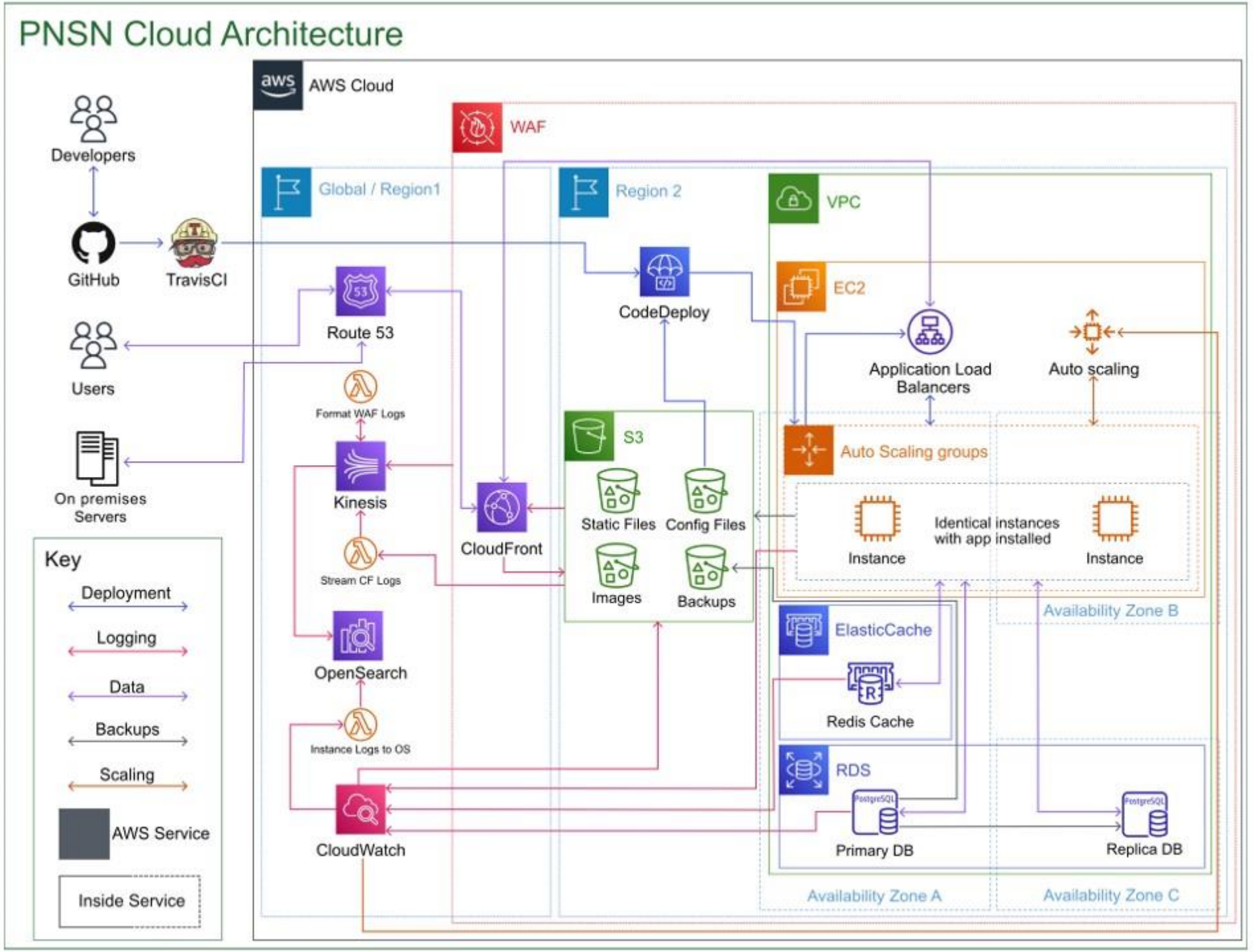


Diagram shows a generalized version of our AWS structure.

PNSN Apps Hosted on AWS:

- pnsn.org
- SQUAC
- ShakeCast

Some Advantages:

- Monitoring tools
- Built in integration between tools
- Ability to quickly scale
- Only paying for what you use
- Physical servers can be anywhere

Some Disadvantages:

- Learning curve
- Many similar cloud services and solutions
- Different skill set than on prem

How to Manage Cloud Infrastructure

- PNSN
  - CodeDeploy
  - CI/CD
  - Monitoring
- EHP
  - Content delivery network
  - Gitlab runners
  - CDK (in) to CloudFormation (out)
- SCSN
  - Terraform

Discussion

Survey question: *Biggest challenges to adopting cloud technology*

- Funding/Lack of human resources
  1. Identifying a set of tasks that can be easily transferred off-premises.
  2. Identifying a suitable cloud service provider to host these tasks.
  3. Learning how to appropriately use the services offered by that cloud service provider.
  4. Remaining up-to-date with that cloud service provider's best practices and recommended services.
  5. Understanding how to avoid vendor lock-in.
  6. Understanding the anticipated costs a priori.
- Finding cloud developers with a true problem solving and exploration mentality
- Extremely limited bandwidth at sites limits ability for dual acquisition

Survey question: *Biggest challenges to adopting cloud technology (cont)*

- ...Adapting mindsets and software to leverage the cloud responsibly is likely our biggest challenge.
- There hasn't been a significant need requiring the cloud
- Making the commitment to move existing processes into the cloud has been a large factor. When services are running on a local server there isn't a need to take the time to make a change to hosting them in the cloud. ... Additionally, there is a preference among our IT team to use servers entirely in our control that we have physical access to.

Survey question: *Hardest lesson learned using the cloud – Hype vs. Reality*

- The auto-scaling of resources in the cloud are too slow to respond to seismic events that could happen at anytime, thus paid-for services need to be standing 24x7 even when idle 'most of the time'.
- ...For AWS specifically, there's quite a few different services that all do similar things and there's a learning curve to figuring out the correct tool to use. When most of the staff is used to the classical approach and don't have any issues with it, there isn't always a case to make for taking the time to learn new services and approaches.
- Some aspects are not as scalable as AWS claimed they were
- Unexpected charges, especially for S3 where even deleting objects has a cost!

Survey question: *Biggest success story using the cloud*

- MyShake app
- Running production AQMS in EC2
- ...From a web development perspective, using cloud services enables us to not have to deal with any hardware issues and to quickly allocate new resources to handle large spikes in traffic... For our public website, in the case of a significant earthquake locally, we have some security in knowing our site is hosted off campus and won't be affected.
- Many of our systems are in production in the cloud. This is timely as our on prem data centers are experiencing more issues such as power and network outages.
- Robust decentralised storage of CSN station data measurements.

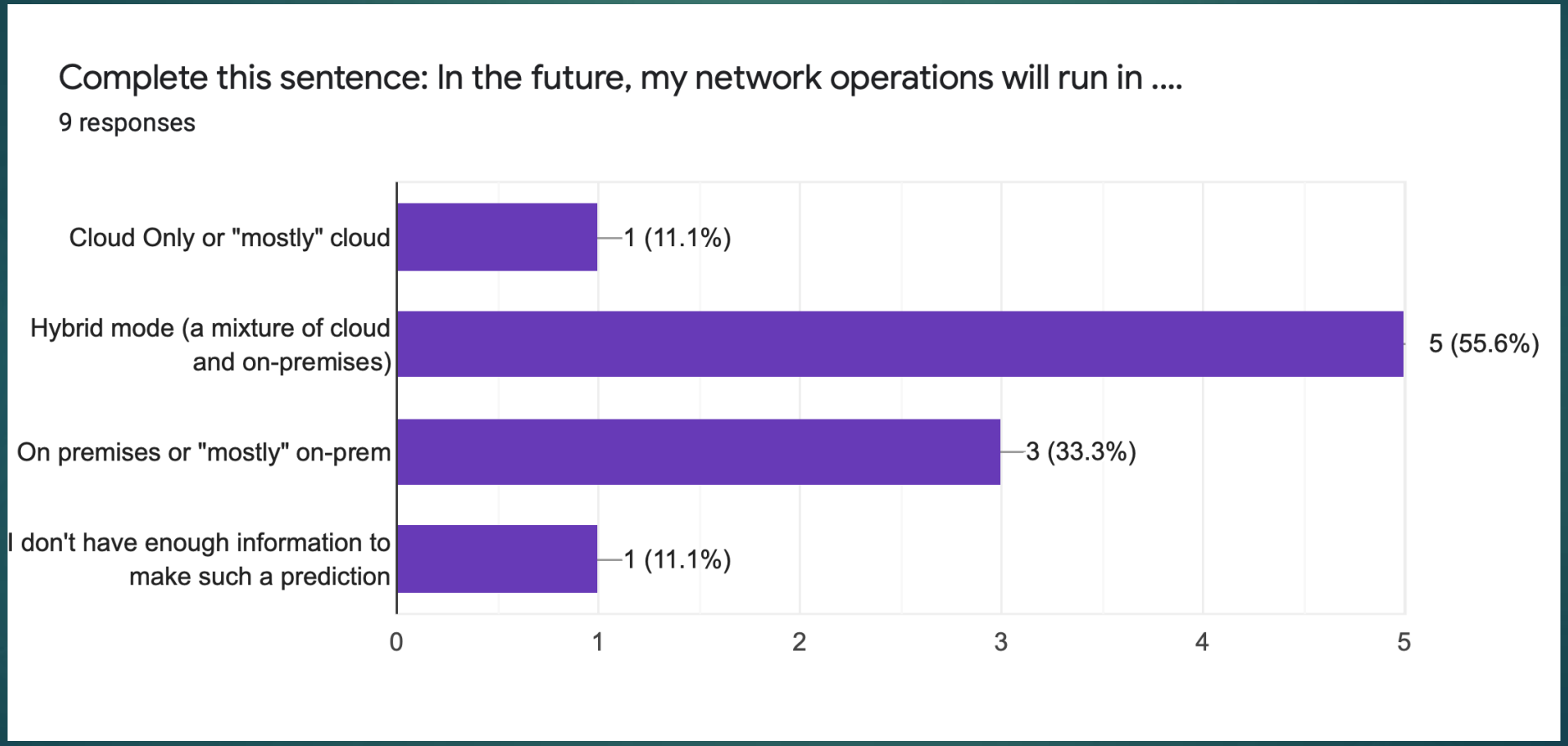
Why use the cloud?

- **Why use the cloud?**
  - Processing high volumes of data (over 500+ stations) in real time. We need the picks to arrive back to their outbound ring in short order
    - If we process the last 30 seconds of waveforms, we want the predicted picks to arrive back within 30 seconds of extracting out those waveforms.
  - Problems when running locally
    - With a high volume of data, we can stress out a single server (such as if done locally). This can progressively lead to longer prediction times over a long, continuous run.
    - Multiprocessing to increase speed in a single server can also lead to out of memory errors, due to running out of CPU threads. This was a problem when running 500 stations at once.
  - Cost considerations
    - Monitor AWS' Cost Explorer for daily or monthly costs based on AWS resources used.
    - Need to experiment to find the right balance in cost (such as comparing using EC2 servers versus lambda functions). AWS provides cost estimators based on time of use.

Advantages of using the cloud

- **Vertical scaling**: Increasing the capacity of a single machine.
  - With AWS, we can always spend more on compute resources.
- **Horizontal scaling**: Multiple servers working as a single logical unit.
  - This is particularly intriguing, because we can adjust to periods of high load (ie such as a major earthquake, when we expect to find more picks and have longer processing times). If we get more stations in the field we can also adjust to that.
  - **Elastic load balancers** can provision more EC2 instances to distribute the load across multiple servers.
  - Can use **serverless applications** (lambda functions, Fargate). Lambda functions can each process a subset of stations, and are easier to provision/implement.
  - We created an **API endpoint** to predict picks on a lambda function, so anyone can POST a single 3-channel station data and receive picks. Highly scalable and can be widely utilized.

Survey question: Cloud only, Hybrid, On-prem?



Other questions

- What will you do next?

Thanks

Rayo Bhadha

Chris Bruton

Julian Bunn

Lynda Lastowka

Kyla Marczewski

Ryan Tam