Panorama

Open edX® analytics reimagined by Aulasneo
Who we are

Esteban Etcheverry
esteban@aulasneo.com

Andrés Gonzalez
andres@aulasneo.com

https://aulasneo.com/en

* Compared to similar systems based on traditional webapp architectures
What is Panorama

• A serverless approach to Open edX® analytics
Why Panorama?

*Thanks to its fully serverless architecture, Panorama is*:

- **Faster to implement**: up in a few minutes.
- **Simpler**: no code, no Open edX code change, no application to install.
- **More scalable**: worldwide integration of multiple Open edX instances.
- **More flexible**: easily integrates with external datasources.
- **More controllable**: granular control to data access.
- **More customizable**: easily create your own reports.
- **Smarter**: use ML to augment data.
- **More durable**: mostly independent of Open edX releases.
- **Safer**: it doesn’t require open ports or APIs for data extraction.
- **More secure**: control access with IAM.
- **More reliable**: based on a serverless architecture.
- **More efficient**: it doesn’t load heavily the Open edX instances.
- **Easier to maintain**: no OS, no dependencies, no infrastructure.
- **Cheaper**: it’s cost model is based on use.

*Compared to similar systems based on traditional webapp architectures*
Auslaneo Panorama is entirely based on AWS serverless components, without any EC2 instance nor relational databases.
Data gathering

- Design principles:
  - Do not touch codebase
  - Do not load CPU or memory
  - Do not use APIs or open ports

- Data comes from three sources in each instance:
  - MySQL: structured student and course data
  - MongoDB: modulestore (course definition)
  - Tracking logs: events

- These sources are queried by simple cronjobs
- Queries are kept simple, in order not to load the db engine. No filters, no joins, no calculations.
- Incremental data is uploaded to S3 buckets
- Data is compressed to reduce bw
- Data is encrypted in S3
Data pre-processing

- Design principles:
  - Keep data as pure as possible, as close to the end user as can be

- Tracking log data needs some pre-processing:
  - Some events are not well formed JSON records
  - URL events are not categorized
  - Course and block ids extraction
  - Event data structure improvement
Data extraction and transformation

- AWS Glue jobs extract and transform data
- Triggers are based on fixed schedules
- Data transformation and ML algorithms can be applied here
Data lake

- Transformed data is stored in a S3 data lake
- Parquet format
  - Huge datasets (Apache Spark, Hive, Hadoop)
  - Semi-structured
  - S3 storage
  - Data is partitioned to improve performance and lower cost
    - Partition on LMS URL allows analytics across instances
    - Partition on date allows processing of tracking log files
- Data catalogs store schema information
- AWS Glue crawlers update the schema and create partitions
Data querying

- Data in Parquet format is queried using AWS Athena
- Queries are in Presto SQL language
- Views can be created with joins and calculated fields
Data presentation

- AWS Quicksight is the presentation engine
- Data sources get data sets from Athena
- SPICE allows fast in-memory joins, filters and calculated fields
- External data sources can be added and joined
- End users can be granted author access
- Dashboards can be embedded in external sites
- Row and column level security allows granular control on user visibility
Some screenshots
Improvement opportunities

- Create new dashboards to align with Insights and other analytics
- Overall optimization
- Improve data ingestion of tracking events
- Embed visuals in third party sites
- Create transformations to aggregate data and deduce new information
- Use ML algorithms to forecast information
- Automate setup with Cloudformation
Thank you

Q & A

info@aulasneo.com