In the end, I was successfully able to aggregate user data for usage analytics on multiple website by writing

multiple json files and folders to an S3 bucket with just one Python file.

CREATE EXTERNAL TABLE IF NOT EXISTS user\_metrics.cloudwatch\_json (

`logStreamName` string,

`timestamp` TIMESTAMP,

`message` struct<

`msg`: string,

`anonymous`: boolean,

`url`: string,

`user`: struct<

`id`: string,

`uuid`: string,

`email`: string,

`orgid`: string,

`packages`: array<string>,

`roles`: array<string>

>,

`response`: struct<

`status`: int,

`content\_type`: string,

`time`: float

>,

`request`: struct<

`ip`: string,

`content\_type`: string,

`method`: string,

`path`: string,

`referer`: string,

`user\_agent`: string,

`cookies`: map<string,string>

>,

`loggername`: string,

`level`: string,

`timestamp`: TIMESTAMP,

`req\_id`: string,

`trace\_id`: string

>,

`ingestionTime` TIMESTAMP,

`eventId` string

)

ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'

WITH SERDEPROPERTIES (

'serialization.format' = '1'

) LOCATION 's3://log-accumulation/2021/08/16/'

TBLPROPERTIES ('has\_encrypted\_data'='false');

SELECT logStreamName, timestamp, message.msg, message.anonymous, message.url, message.user, message.response, message.request, message.loggername, message.level, message.timestamp, message.req\_id, message.trace\_id, ingestionTime, eventId FROM "user\_metrics"."cloudwatch\_json" limit 10;

//SELECT message.user.id, message.user.uuid, message.user.email, message.user.orgid, message.user.packages, message.user.roles from "user\_metrics"."cloudwatch\_json" limit 10;

//SELECT \* from user\_metrics.cloudwatch\_json