Hello / Olá / Buenas / Merhaba / やあ / 你好 / Namaste !

You have successfully passed the first stage of our selection process: congratulations and good luck!

The challenge

Your task is to create a data lake with the events coming from a Kinesis stream.

The stream delivers around 1M events/hour and there are 100 different types of events, all mixed together. Events are json objects. All of them contain the common fields:

- `event_uuid` - unique identifier of the event
- `event_name` - string identifying the type of the event, it can consist of multiple parts separated by ":", for example: "account:created", "lesson:started", “payment:order:completed”
- `created_at` - Unix timestamp of the event creation

The rest of the payload consists of fields related to the event type.

The saved events should have the following, additional fields:

- `created_datetime` - date and time from the `created_at` field, in the ISO 8601 format
- `event_type` - the first element from the `event_name` field
- `event_subtype` - the second element from the `event_name` field

Please assume that the system will be working on the AWS cloud.

Design questions

- How would you handle duplicate events?
- How would you partition the data to ensure good querying performance and scalability?
- What format would you use to store the data?
- How would you test the different components of your proposed architecture?
- How would you ensure the architecture deployed can be replicable across environments?
- Would your proposed solution still be the same if the amount of events is 1000 times smaller or bigger?
- Would your proposed solution still be the same if adding fields / transforming the data is no longer needed?

Deliverables

- A Makefile automating environment creation and test execution of your proposed solution.
    - When provisioning infrastructure, please use Terraform.
    - When programming, please use Python 3.
    - Script creating data structures (if needed)
- A README file documenting your solution:
    - Architecture diagram.
    - Short explanations of the technologies chosen and why
    - Answers to the design questions.