2023-05-08 Architecture WG Meeting Notes

Date

08 May 2023

ZOOM Meeting Information:

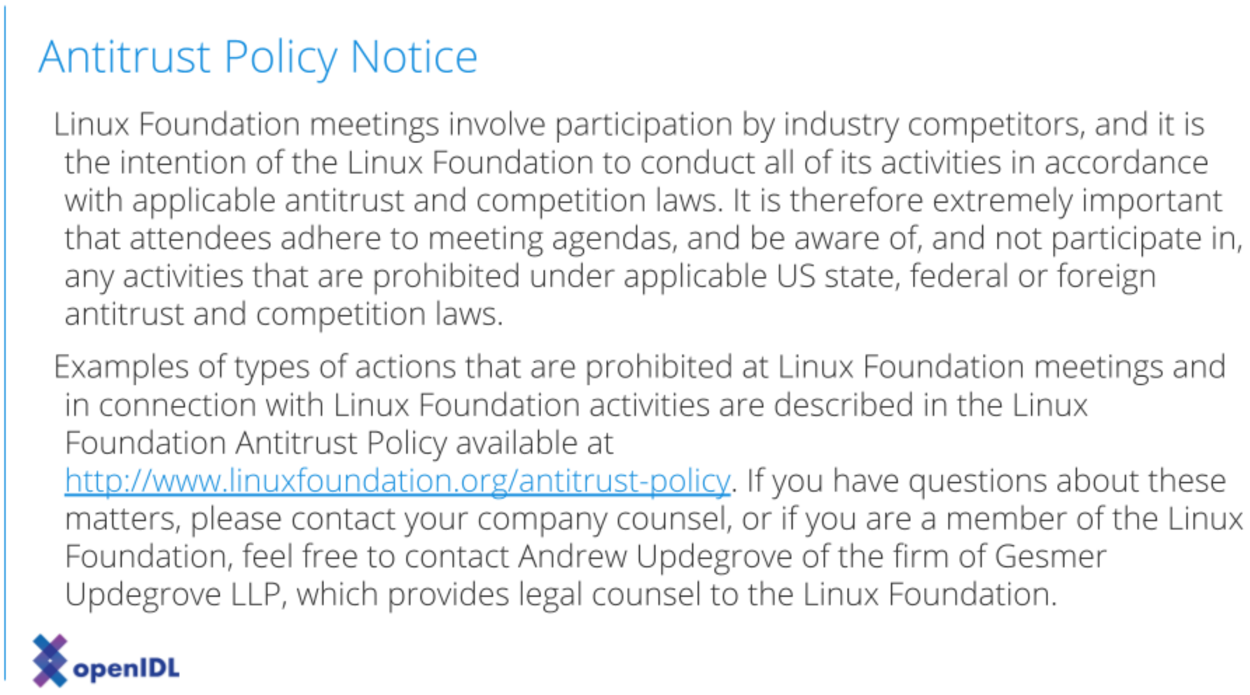
Monday, May 8, 2023, at 11:30am PT/2:30pm ET.

GMT20230508-1...1920x1080.mp4

Join Zoom Meeting

https://zoom.us/j/7904999331

Meeting ID: 790 499 9331



Attendees:

- Sean Bohan (openIDL)

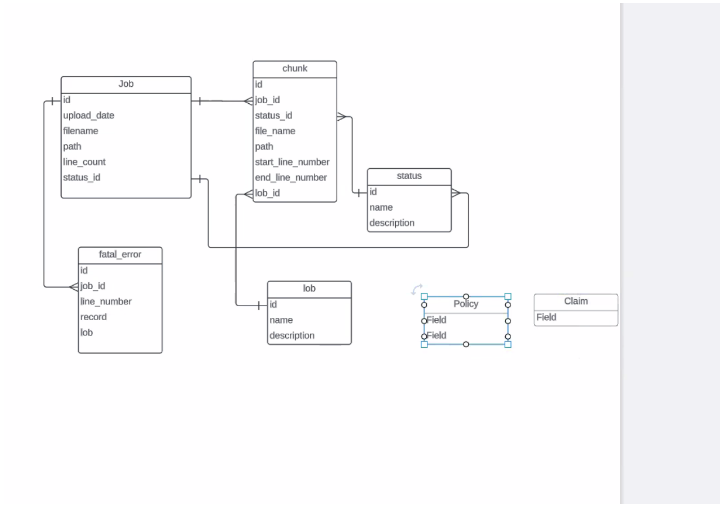
- Mason Wagoner (AAIS)
- David Reale (Travelers)
- Joseph Nibert (AAIS)
- Dale Harris (Travelers)
- Peter Antley (AAIS)
- Ken Sayers (AAIS)
- Tsvetan Georgiev (Senofi)
- Ash Naik (AAIS)
- Brian Mills (AAIS)
- Yanko Zhelyazkov (Senofi)
- Adnan Choudhury (Chainyard)
- Faheem Zakaria (Hanover)
- Allen Thompson (Hanover)

Agenda:

- MS Hurricane Zeta POC Architecture Discussion (KenS)
- Update on openIDL Testnet (Jeff Braswell)
- IWG update (YankoZ)
- Update on RRDMWG and internal Stat Reporting with openIDL (Peter Antley)
    - OLGA: Implementation Discussion
- AOB:
- Future Topics:

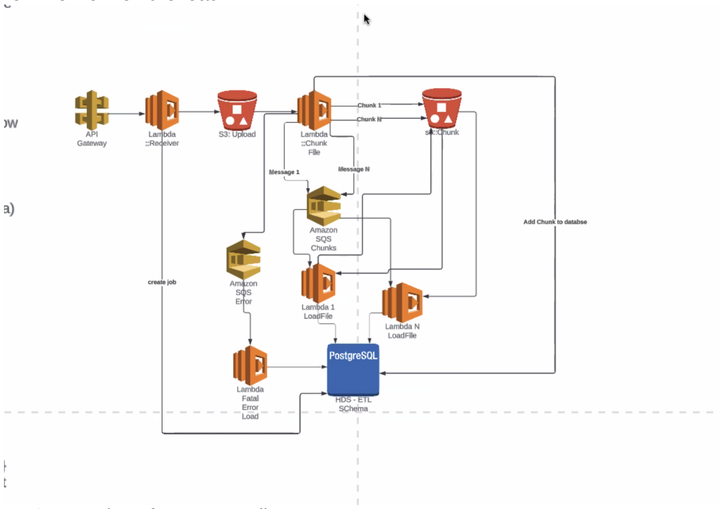
Notes:

- Hurricane Zeta
    - initial planning
- Testnet
    - deleting AWG for May 29
- DMWG
    - had first Weds meeting last Weds
    - meeting this Weds
    - giving actuary team points of interest in terms of stat plans
- OLGA - replacing SDMA
    - talking with biz stakeholders about inline editing, bulk editing, when and why using bulk editing
    - policy and claim tables

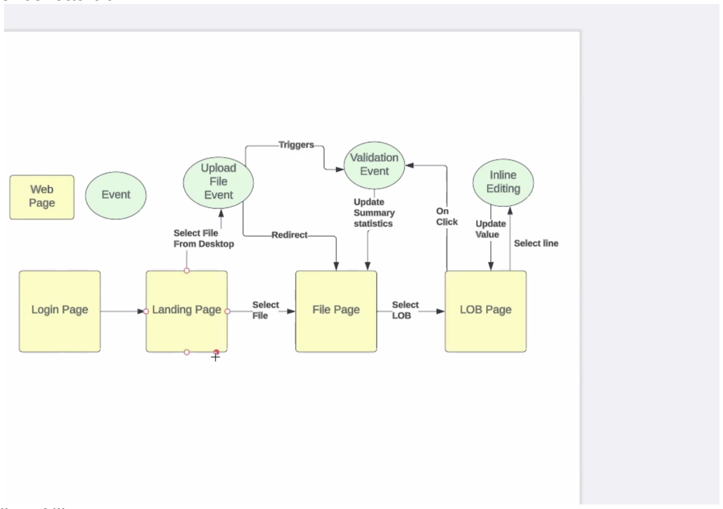


    - allow people to submit large docs into system and process large docs
    - make system process files up to 5gb
    - if file greater than, suggest they make 2 files
    - JN - records? thinks 5gb is 30-35MM files depending on data
    - raw (not zipped or unzipped?)
    - KS - number of records in an individual file
    - 5gb limit - 30MM records, over the limit of what they are shooting for
    - taking one 5gb file, breaking into chunks
    - doing the chunking, want to break chunks up into two ways
    - break up so no greater than 10k records, so that chunks only have records for 1 LOB
    - multi-line file, break into mult threads, diff data objects, for HO and PA and CA, diff chunk for each record
    - work to simplify how we look up records, track job and get rid of data
    - what if a record cant be loaded?
    - want to put the bad record into a fatal error table: job ID, line # and raw string value
    - fatal error - record doesn't match right schema
    - run thru a job, fail to load still make accessible

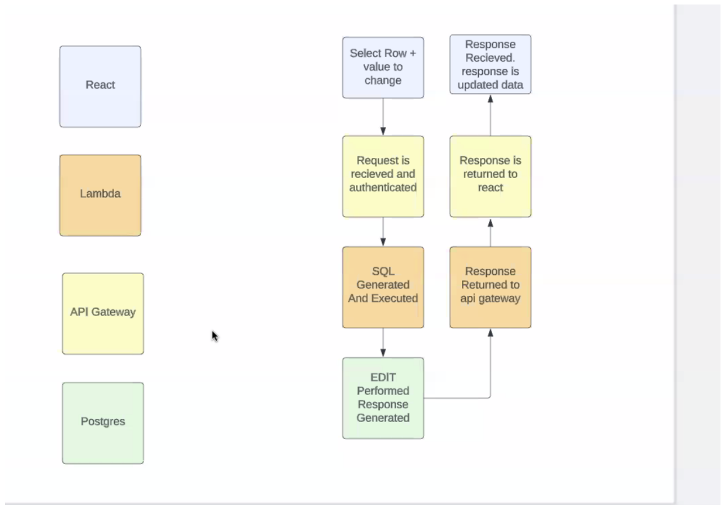
- status with chunks and status of job and internal state based on 2 status IDs
- info on the chunk it the most recent
- how we work on the load



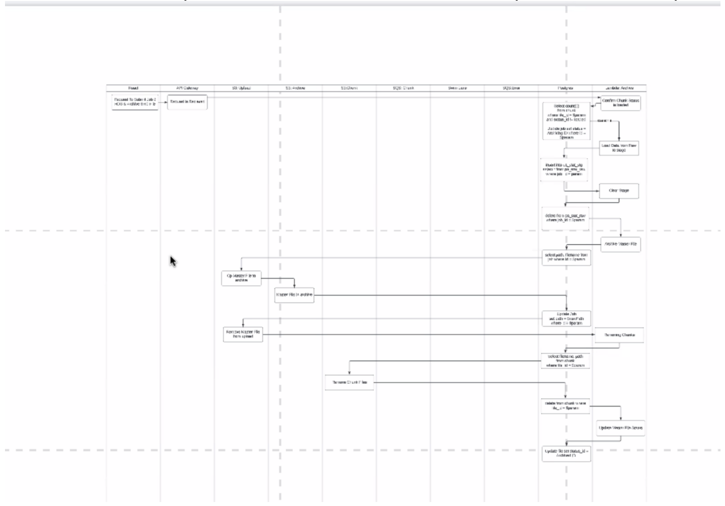
-
- AWS an various features avail
- Elastic File Service - attach more storage to lambda than have available to me
- doesn't know exact date - march? - can now partiition up to 10GB of data with a lambda
- 5gb file? can use lambda with normal ephemeral storage to do all chunking
- no EFS
- save from complexity
- flow:
  - api gateway receives file
  - lmbda catches from gateway
  - registers job with postgres
  - into S3 bucket
  - another lambda starts put event
  - opens file
  - read row thru row
  - validate row matches existing schema, will allow to load correctly
  - chunks based on LLBs
  - write to S3 bucket
  - after chunk file written to bucket lambda writes to queue service with metadata about chunk
  - more lambdas
  - watching queue
  - as items pile up in queue, lambdas pick up files, from s3, load into postgres
  - row that doesn't meet proper schemas, writes to error queue
  - how loading db
  - once load db



  -
  - list of llbs present
  - see specific data
  - perform inline editing there
  - in terms of uploading, next step validations
  - (more on validations next week)
  - upload file, run validation, do inline editing


React
Lambda
API Gateway
Postgres
Select Row + value to change
Request is recieved and authenticated
SQL Generated And Executed
EDIT Performed Response Generated
Response Returned to api gateway
Response is returned to react
Response Recieved. response is updated data

-
- react app - providing back end services + unique ID for a row, value change and value itself
- api gateway receives request and thru cognito auth request
- lambda takes info sent, combo of value row and column, gen sql, connect to db and perform that edit
- after edit - reselect updated row, use lambda to return it, api will be able to update UI with data from back end



-
- resources across the top
- upload file, ran validations, made corrections, validated again and now ready for submission
- most is db related
- Amazon lambda toolkit called "Powertools"
- others recommend other tools?
- XRay?
- highly observable applications
- FZ - formatting could be easier to define, use any tool to read log files
- GRAYLOG - initial fields always fixed and present, var fields as appropriate - formatting style
- too far down AWS might not have anythign on azure side
- cloudwatch
- similar capability on the azure side
- FZ - containerize as much as possible, w/o leaning on a specific clouds capabilities
- optimizations for azure and aws
- cloud agnosticism
- deploiy to diff clouds
- HL is a kubernetes based service
- dockerized vs kubernetes
- concern - may need to reimplement all from azure perspective
- react apps - same API routes, react apps would be similar
- saving a lot of implementing that AWS does for you (lot of management in managed services
- larger group discussion - open it up for awareness
- containerize to make it more cloud agnostic
- dont need sophistication of kubernetes
- multi-cloud an issue
- lots of use
- offering as SaaS
- make it multicloud or overcomplication as Kub hard to afford

- a lot for not a lot of gain
- loking at containerizeing as much as possible, deploy on diff clouds
- service in azure to deploy w/o kub (so as AWS)
- once redockerizing/recontainerizing things - this is on demand workflow - fits well for serverless - containerize lose features you like, cloud less attractive
- sometimes opposite
- cloud and cloud native makes sense
- sparse and infrequest
- do an architecture decision, get two options out there - side by side, make a call
- move forward, knows way he is leaning, bring something up
- agenda for next week - AD
- when to use Kub? Reference implementation, for OLGA, some kind of containers/containerbased, or baremetaled
- limited scope of OLGA
- ,

Time	Item	Who	Notes

Documentation:

Notes: (Notes taken live in Requirements document)

Recording: