

2023-03-30 Infrastructure Working Group Agenda and Meeting Notes

March 30, 2023

30 Mar 2023

Antitrust Policy Notice

Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at <http://www.linuxfoundation.org/antitrust-policy>. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrave of the firm of Gesmer Updegrave LLP, which provides legal counsel to the Linux Foundation.



When:

Tuesday, October 4, 2022

9am to 10am PDT/12:00PM-1PM EST

ZOOM Please Register in advance for this meeting:

https://zoom.us/join/zoom/register/tJwpcO2spjkpE9a1HXBeyBxz7TM_Dvo8Ne8j

Attendees:

- Sean Bohan (openIDL)
- Nathan Southern (openIDL)
- Peter Antley (AAIS)
- Ken Sayers (AAIS)
- Surya Lanka (Chainyard)
- Yanko Zhelyazkov (Senofi)
- Allen Thompson (Hanover)
- Adnan Choudhury (Chainyard)
- Aashish Shrestha (Chainyard)
- Jeff Braswell (openIDL)
- Faheem Zakaria (Hanover)

Agenda Items:

- Antitrust
- ND POC: Learnings, What Worked, What Didn't, What Was Changed
- Next topics
- AOB

Minutes:

- openDL ND POC Changes
 - 3 - Infrastructure, Fabric, Application POV
- Infrastructure
 - added code to deploy the lambda function for report processing and upload UI
 - S3 bucket for cloud distro and AWS cert manager
 - upgraded Kub to 1.23 (1.22 was getting deprecated)
 - test environ updated
 - added cert manager for ssl certs for all endpoints
 - updated self-managed node groups to EKS managed node groups
 - self managed was using launch configs, getting deprecated in Dec - had to update them to EKS manager node group (template not configs)
 - Added jenkins pipeline to deploy upload UI app config
 - Upload UI was part of ND POC
- KS
- some things specific to ND, may take it out of base code and look at it as reference
- Surya
 - Fabric
 - in HLF deployment, upgraded the images from 2.2.3 to 2.2.9
 - have the bug fixes as part of 2.2.9
 - fixed the chaincode to run after start
 - whenever there is a peer restart, need to upgrade chaincode to make it work
 - there is a restart of the peer, comes back up, should / will be able to create chaincode w/o issues
 - fixed
 - not related to ND POC, but in general
 - Application
 - ND POC - need to deploy incidence manager to analytics node
 - with changes done, deployed data manager to Analytics node
 - new params in application component configs
 - automated gen of app component files through ansible templates
 - ansible app config files and load into one based on node type
 - integrated through jenkins pipeline
 - during node setup or new changes, can just trigger pipeline to update add app config files to the cluster
- Why upgrade was performed? Specific bugs? Or just in case and what problems occurred
- Surya
 - w/ respect to bugs, necessary to keep code up to date, using 2.2.3 v of Fabric after which there is .4-.10
 - keep code up to date w/ respect to 2.2 vers
- JB
 - was latest vers of fabric related to smaller max PDC size?
- Surya - already there due to couch.db
 - 2.5 has support,
- Adnan
 - re: DC size, Fab has default value, going to static, not configurable
 - do not want to go over max
 - larger dataset in PDC reduces performance due to larger transactions
- Aashish
 - couchdb 4, planning to make set limit (8mb)
 - future versions would encounter problems
- KenS - concern about latest ver?
- Yanko - target specific fixes, cautious
 - new vers may intro some other issues
 - wondering what specific bugs
- KS - we dont want to get 2/3 vers behind, takes imporrible lift, stay within certain # of fabric
- YZ - minor fixes and features
 - major vers require regression testing, etc
- AS - did not update minor vers, just latest patch vers
- KS 2.2.3 - 2.2.9
- AS - next would be 2.4, feature wanted was to deleted PDC by triggering functions is in 2.5
- KS - carriers dont want data in PDC
- results of extractions, once in report, should be in PDC
- KS - didn't test for some sizing
 - some unnecessaary logging
 - of things to do inside abd required beefing up machines
 - ND is an outlier - passing 100k rows, diff than stat reporting
 - more than running out of memory issues, probs with timeouts and restarting pods, etc.
- Aashish
 - haven't touched performance tests
 - code optimization around loops too
- Adnan
 - most taken care of by resource restructuring
 - specifically when running EP in Mongo, resources not enough
 - after beefing up resources, some loops and logs needed to be reprogrammed
 - longer timeouts, etc.
 - 1MM rows of data, give proper time, etc.
- KS
 - had to batch stuff in PDC
- AC
 - going over config value for PDCs

- chopped up w/ config value and saved results in PDC 1 by 1
 - taken for further processing, next set of process
- YZ
 - perf problems due to chaincode and how it processed data
- AC
 - unique nature of data itself, had to recalibrate a few things
 - comparison of data needed to be efficient
- YZ
 - report processor - report being created
 - what are the problems on the network side - fix or extra effort to make it work
- AC
 - saw timeouts
 - tranax timeouts
 - needed where we were trying to inc the resources
 - made sure the node with peer was not overloaded
- SL
 - doc memory issue
 - processing larger data in PDC, couchDB issues with peers (size issue)
 - some app components getting killed due to "out of memory"
 - mult cases where we see the processing time is taking longer and getting killed due to OOM
- YZ
 - analysis on why running out of memory?
- AC - found issue
 - openIDL has 1 status DB, saving status of datacall
 - for some reason fails, scheduler comes in and tries to finish the job, comes back every 30 or 45 min
 - in test environment
 - some data calls did not - saw test environment was doing transax even though not doing data calls
- YZ - issue was application side, not fabric
- JB
 - Resource sizing for HDS in AWS is distinct from resource requirements for Fabric nodes, is it not ?
 - If HDS AWS resource for carrier is different than the Fabric node resource, the Fabric node resource would not need to be so large ?
 - diff resource sizing reqs were for app requirements, not the node itself
- AS - typical data call, size of data less than ND?
- KS - EP would be hundreds of records, small results at extraction
- PA - we will be somewhere between, comes back as JSON, formatting data layer, string of JSONs
 - under 1000 element JSON
- AS - will cut down a lot of processing
- KS - stat reports are similar, will have other situations
 - MS POC might have a similr result set
- PA - much bigger, not drivers, whole state
- AS # of carriers? per state?
- PA - 100+ sep entities reco by NAIC
 - 200 by AAIS
 - less than 3k total
- KS - way expecting to see it unfold, loading on behalf of carriers into multitenant node
- PA - load testing to see how much we can fit
 - lots of carriers in a single node
 - 200+ carriers
 - a lot of small mutuals, crrier node 4 has 100 carriers in same table
 - data call, agg info, wont have all the primary keys
- KS - ind nodes for less than 20 carriers for a while
- AS - multi-tenant node size - MS ?
- KS stat reporting will stretch multitenant
- PA - didn't have any of these carriers working with us
- KS - putting data in was a huge win
- KS - cool thing
 - adding HDS to analutics node made it possible
 - wasnt considered in the original design
 - allowed DOT to load data that could be used by reporting processor
 - ability to quicly pivot and create diff reports was a big win as well
 - data quality issues
 - bad VIN situation, vehilces not on the list due to not insurable
- KS - how do we merge?
- YZ - tickets in github with problem and solution, how approached, get aproved, can/has addressed can create a branch to get approved and merged
- KS
 - how do we decide
- YZ - focus on application side issues, few probs
 - processing data, etc.
 - Network side - going with operator so we prob wont use those changes, not relevant to new openIDL
 - ex: performance prob with processing data during data call, title of a git hub issue
- AS - can move over things on trello, tracking this, as issues in github
- JB - any reason to make a repo for ND app specific stuff, baseline concept
- KS - are we refactoring a repo? roadmap, backlog
- YZ - merge the fixes into MAIN, then refactoring later
- KS - grab Trello items, pick what applies, merge and refactor
- PA - Mason and PA working on ETL, are we making a new repo? stay where we are?
- KS - ETL will stick around - kee working there



GMT20230330-1..._1440x876.mp4

Action items:

-
-
-
-