2023-04-13 Infrastructure Working Group Agenda and Meeting Notes

April 13, 2023

13 Apr 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 Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

openIDL

When:
Tuesday, October 4, 2022
9am to 10am PDT/12:00PM-1PM EST
ZOOM Please Register in advance for this meeting:
https://zoom.us/meeting/register/tJwpcO2spjkpE9a1HXBeyBxz7TM_Dvo8Ne8j

Attendees:

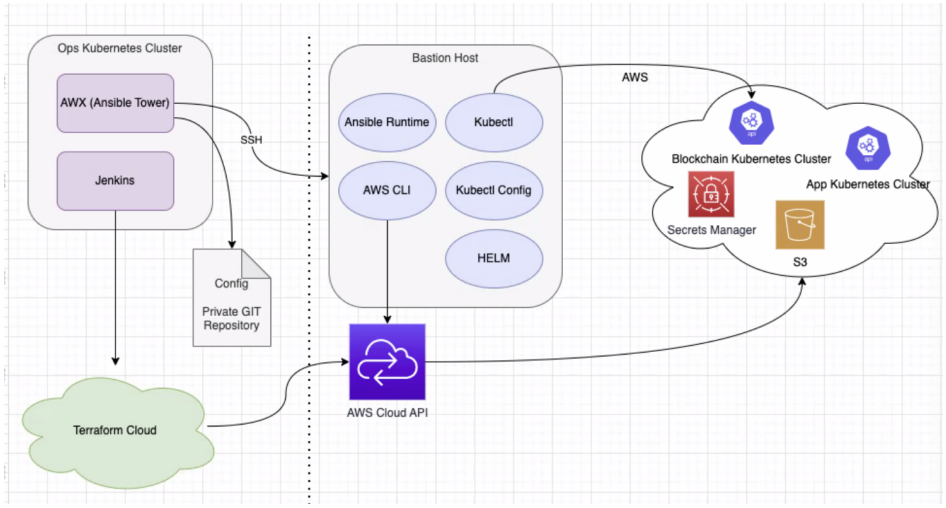
- Sean Bohan (openIDL)
- Yanko Zhelyazkov (Senofi)
- Jeff Braswell (openIDL)
- Ash Naik (AAIS)
- Faheem Zakaria (Hanover)
- Peter Antley (AAIS)
- Surya Lanka (Chainyard)
- Ken Sayers (AAIS)
- Mohan (Hartford)
- Allen Thompson (Hanover)

Agenda Items:

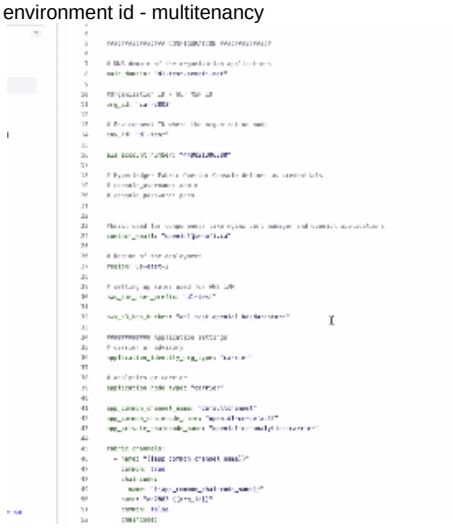
Architectural Decisions

Minutes:

- go high level on the current architecture and how things work with operator
- focus on the ways we set up infra and installing AWS components
- focus: how we install the openIDL as a node



-
- left side operations tools
- right side openIDL node itself
- comparison from before and now
- components exist on a single account and could exist on sep accounts
- always one ops cluster per node?
- depends on how network is structured
- for IP, may be managing mult accounts at same time, may not make sense to have N number of ops clusters
- ansible and awx have multi tenant approach
- IP may manage nodes A, B, C, to minimize costs, dont need AWX on every account
- Privacy of whats on the cluster a dependency
- flexible with approach
- every provider or carrier has own requirements
- large carriers - security is important
- can people see if there are creds on a multi-tenant ops cluster
- doing all from single cluster, all managed from one place
- make sure it works the other way
- asked by big cos
- question of deployment
- creds in AWX are encrypted
- every org setting up, dedicated users and orgs
- whoever is in control of that user from AWX perspective is the only one with access (one way)
- like having a token
- accounts and creds in aws, not sure how implemented or consume as a service, most funct is multi-teneant on a logical level than pure DB level
- normal to have part of your data, account sitting somewhere on dedicated shared DB, sep by encryption, whatnot
- purely for flexibility
- every carrier can decide what they have in their infra
- nothing against that
- every will make own decisions, logical separation - security is high priority
- Ansible is pretty secure in that sense, still concerns can do more research
- not separating a carrier hosting a node vs carrier not hosting
- if smaller carriers were to use services of an IP to manage network, an IP could do that from a single ops cluster
- Jenkins - CI/CD
- some UI that can start certain actions, creating infrastructure, setting up AWS account (kub, etc.)
- have jenkins bundleing the infra as code (terraform scripts), uploading, starting the terraform script over from cloud
- works same as before, no major diffs, usually have in cloud, environment variables, pointing to AWS acount, access key, config aspects of kub clusters
- uses terraform IasCode, AWS cloud APIs, first step of the process, major diff, not installing nginex servers, moved nginex tool to create ingress in Kub, functionality delegated to ansible
- ansible - egress point for Kub cluster?
- tool, declarative lang
- allows us to config machines
- usually thru SSH
- oriented towards AWS, thoughts towards Azure
- not trying to make this cloud agnostic, implement best practices in each cloud
- tried to be cloud agnostic for ansible playbooks
- minimize what is specific to cloud
- moved creation of ingress controllers
- want to be able to support mult nodes on same cluster
- exp for cases with testnet or dev net
- many things done to enable multitenancy on same cluster
- ops cluster, HLF cluster
- devnet - exactly that setup, mult nodes running on same Kub cluster and then mult instances
- support diff domain names
- within same account and kub cluster, removes restrictions
- names MSPs needs to be
- more flex
- domain name of a node

- upcoming training workshops - first set of sessions is to set up infra
- what to cover in first sessions
- in terms of workshop - assume it starts with AWS
- other major change - confg repo
- previous implementation, going into same git repo
- all the ansible playbooks
- one repo
- flexible enough
- can have every branch diff carrier
- or sep git repo
- one of the carriers
- can pick the
- in config file - dont push passwords - all creds in awx
- mainly try to simplify the config in order to deploy fabric and openIDL apps
- bunch of configs like AWS account #, org ID, main domain name
- no need for a script - most things are templated already
- can overwrite whatever you like
- environment id - multitenancy
- 
- can be config'd manually
- editing, knowing what/what not to change is important, may be useful to ID elements that need to be supplied
- idea to make it easier, minimize errors
- intermediate step
- documented inside
- another change, spawning and configuring clusters - dont use Kub jobs anymore
- everything that needs to be run and configed is based on Bastion Host
- also using AWS directly, config certain aspects of AWS for running servers
- ex: creation of DNS entries and so forth
- have own terraform scripts for azure
- write extensions to current playbooks
- mod to bastion host
- want to use cloud provider APIs
- at what point does Azure become a part of the decision to use openIDL? now? after you use an aws thing hosted by LF?
- business folks need to weigh in
- from cloud perspective, they partnered with Azure, their chosen cloud provider
- kickoff and running
- not fair to burden project
- one approach - purpose of training, exercise, get hands on
- terraform is a positive for adoption to azure
- an

Action items:

- [ ]
- [ ]
- [ ]
- [ ]