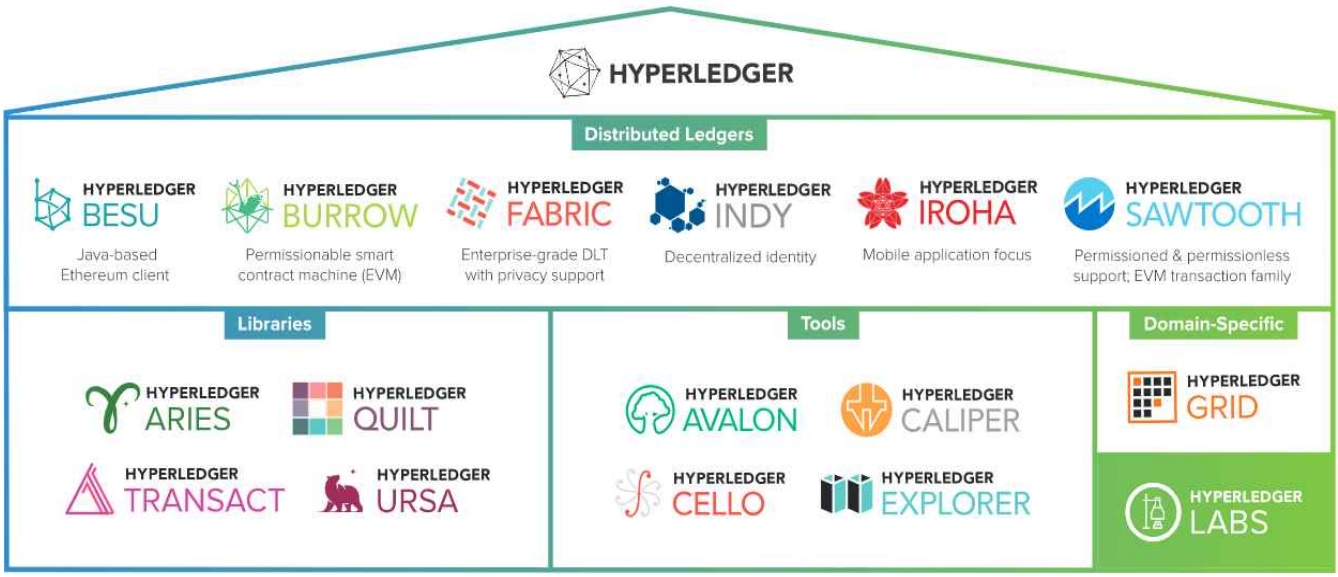
< Hyperledger Meetup #21 >

1. 개요(Hyperledger Greenhouse 소개)

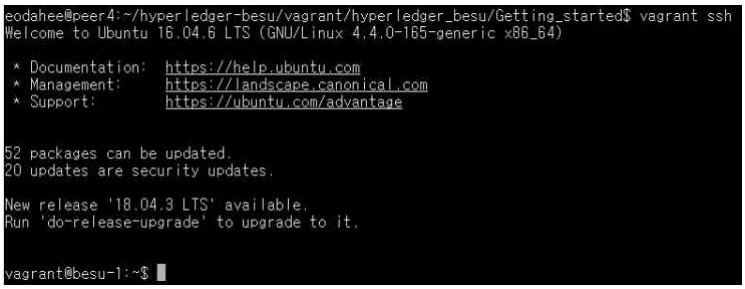


구 분	서비스	내 용
Service	BURROW	EVM 머신
	IROHA	모바일에 적합한 블록체인
	SAWTOOTH	IoT에 적합한 블록체인
Tools	CALIPER	성능 측정을 위한 tool
Libraries	QUILT	블록체인 간 상호호환성을 가져가기 위해 사용

2. Besu 소개
   - 일번어로 base/foundation 의 의미
   - 기본적으로 이더리움의 속성을 모두 갖춤
   - JAVA로 개발(JDK/JRE 11+ 버전)
   - Enterprise + Ethereum + Open source +
   - 이더리움이지만, Hyperledger이기에 접근권한의 기능을 가짐(permission)
     → node permission, account permission
   - Privacy(관련된 참가자 사이의 거래를 비공개로 유지할 수 있는 기능, 특정 노드에 허가된 참가자들만 tx를 발행 할 수 있음, Private Transaction Manager를 통해 Privacy 구현, Orion이란 이름으로 지원)
   - Private Transaction Manager = 이더리움 노드 + 오라이온 노드
   - Privacy Group = Channel 기능으로 지원

3. Besu 실습

- https://github.com/hlkug/meetup/tree/master/000000/vagrant/hyperledger_besu/Getting_started
- Vagrant는 DevOps의 하나로 보고 코드 기반의 IaaS를 제공하는 툴
- Private Transaction Manager를 사용하기 위해 orion을 설치해야 하는데, 이 orion을 쓰기 위해서는 libsodium(crypto 관련)라는 라이브러리가 설치되어있어야 함.
- IBFT 2.0 실습
- 코딩 순서

관련 프로그램 및 라이브러리 설치	$ vagrant up
vagrant로 접속	$ vagrant ssh 
관리자 권한으로 접속	$ sudo su -  (ls 쳐보면 실습을 위한 shell script file 2개 존재)
4개의 노드가 생성되고, genesis 블록이 생성되며 기본 network 설정	# cd libsodium-stable/ # ./configIBFT.sh
./startIBFTNetwork.sh 내용	# ./startIBFTNetwork.sh 1번 Node의 log file을 보면 Encode URL enode://~~~ 가 있음 → 이 정보를 copy → 2~4번 Node의 —Bootnode 옵션에 해당 주소를 넣어줌 → 각각의 노드가 연결되고 서로 통신 가능 → smart contract 사용 가능

- besu network에 노드 4개를 설치하고 각각의 data 밑에 pub, pri key를 배치
- ibft.json 파일을 바탕으로 genesis.block을 생성하고 이를 통해 node에 필요한 key들을 통제

4. 질문사항
- kafka보다 raft가 30% 성능이 더 좋은 것 같다
- 00회사에서 김포와 울산에 지역화폐를 BaaS로 하고 있음
- 국책사업에서도 DID 관련 내용을 선발하고 확대하고자 하는 중
- Token 관련 소문? Hyperledger 커뮤니티에 요구사항이 없어서 2.0에서 뺐다..
  Linux Foundation 소속 기업들이 요청하지 않아서
  각각의 기업들이 Token을 자체 개발 중이다..(IBM도 준비중)
- 내년 프로젝트에는 디지털 자산화에 대한 내용도 나올 것이다.
- 실제 production에서는 CouchDB를 쓰지 않고 LevelDB를 사용
- 배치라이트를 할 때 tx가 여러 번 생성되면 key가 여러 번 호출되고, Fabric에 준비된 것이 있는가? 없음(동일한 key에 대해 read, write에 대한 set이 발생 할 수 있음)
- FabToken을 쓰고 싶으면, JP모건 쿼럼, R3 Corda를 쓰거나 이더리움의 ERC-20, ERC-711 등 엔터프라이즈 블록체인의 코인을 쓰는 것을 추천
- 가능한 core를 건드리지 않았으면 하는 개인의 의견
  오픈소스인만큼 node를 그대로 가지고 와야하는데, 그걸 수정하면 안 될 것 같음
  core를 수정해서 자체 체인을 만들기 보다는 시대의 흐름을 읽어 적합한 블록체인을 찾아 hybrid로 묶는 것 추천
- channel에 신규 org가 들어와야 하는데 한 org에서 sign을 안해준다면? 이런 고민 필요!
  운영사항의 확장을 고민해야 할 시기가 2020년이지 않을까 생각한다.
- DID만으로는 business model이 없음 → 이 모델을 고민 할 필요 있음
- 우리나라가 블록체인이 잘 안 되는 이유? 성격이 급해서
  PoC를 하고 당장 눈에 보이는 성과가 없다면 서비스를 중지해버림
  외국은 먼저 컨소시엄을 구성 → 각 entity들이 얻을 수 있는 이득을 계산 →
  프로토타입을 개발 → 점차 enhance 시켜가며 발전
- 보통의 컨소시엄에서 나오는 TPS는 200정도
- 성능 향상을 위해서 peer와 chaincode 사이를 건드리기보다는 HFC를 건드리는 것이 좋을 것 같음(=의미 없음 / 보지 말아라)
- 삼성의 넥스레져는 peer 앞단의 연결을 통제해 8000 TPS를 찍음
- Fabric 1.2나 1.3에서는 동적인 peer나 org 추가가 불가능했는데, 1.4에서는 가능한 여러 가지 재료가 나왔기 때문에 그런 부분에 대해 고민해 볼 필요가 있음
- Fabric은 합의가 없어서 tx가 발생 할 때 블록이 생성되지만,
  IBFT는(3F+1) 최소 4개의 Verifier로부터 시작함(계속 블록을 생성해서 1분 지나면 100MB가 쌓이는 단점은 있음)
- Fabric의 단점은 p2p로 하고 싶어도 채널을 생성해야 함(장점이자 단점)
  2개의 node일 때는 상관이 없지만, 많아질수록 matrix로 늘어나기 때문에 어마어마함
  10개의 org가 있으면 gossiping이 일어나게 됨 → 결국 죽어버림..
- Private Data를 위해서 2개의 section이 있다고 생각하면 됨
  A가 B에게 private하게 데이터를 공유하고자 한다면? privateFor라는 옵션값에 B의 Pub_key값을 넣고 보내면 B가 열어볼 수 있음(다만, 이 때 RDB를 사용함)
  Besu의 장점은 Node가 추가될 때 방화벽만 열려있다면 바로 들어 올 수 있음
- 최근 쿼럼은 domain으로 접속이 가능해짐(IP로 접속 가능하다는 건 대단한 것)
- core를 수정하면 거기에 묶여버리기 때문에 아직 블록체인이 안정화되지 않은 시점에서는 다양한 블록체인을 접해보는 것이 더 좋을 것 같다는 생각.