Mission Control Indexer Workshop

Documentation

https://github.com/graphprotocol/mission-control-indexer

Phase 0

Phase 0

The Mission

Deadline:
August 12, 2020 EOD

Results:
August 17th, 2020

1. Deploy initial infrastructure
2. Index selected subgraphs
3. Share endpoints with The Graph
4. Serve metrics via Prometheus
5. Serve 10 queries/second with less than 0.05% error rate for queries

Phase 0

Infrastructure

1. Graph Node
2. Postgres (version 12+)
3. Prometheus

+ Ethereum node or provider

Phase 0

Subgraphs

Moloch

Mainnet (archive node)

1000 entities, indexes in a few hours

Synthetix (bonus points)

Mainnet (archive, traces)

1.6M entities, indexes in a few days

Uniswap

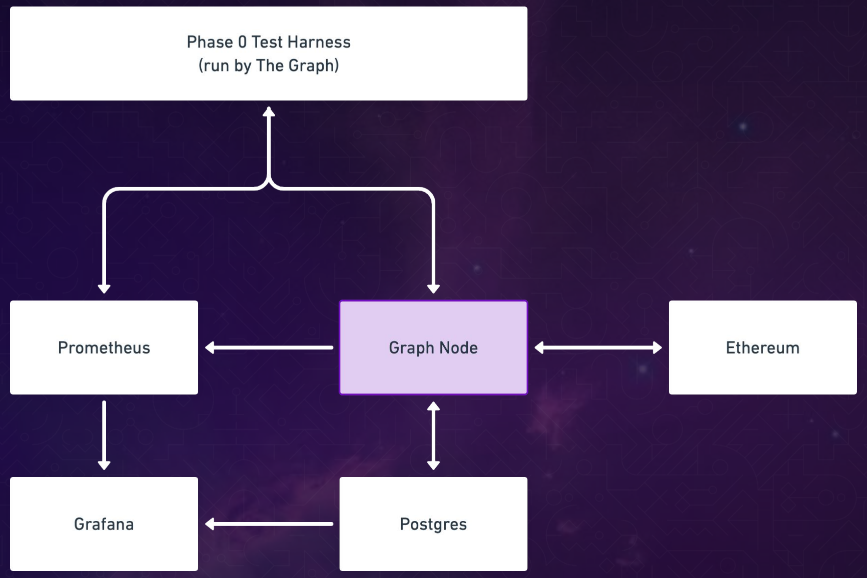
Mainnet (archive node)

4M entities, indexes in a few days

Infrastructure

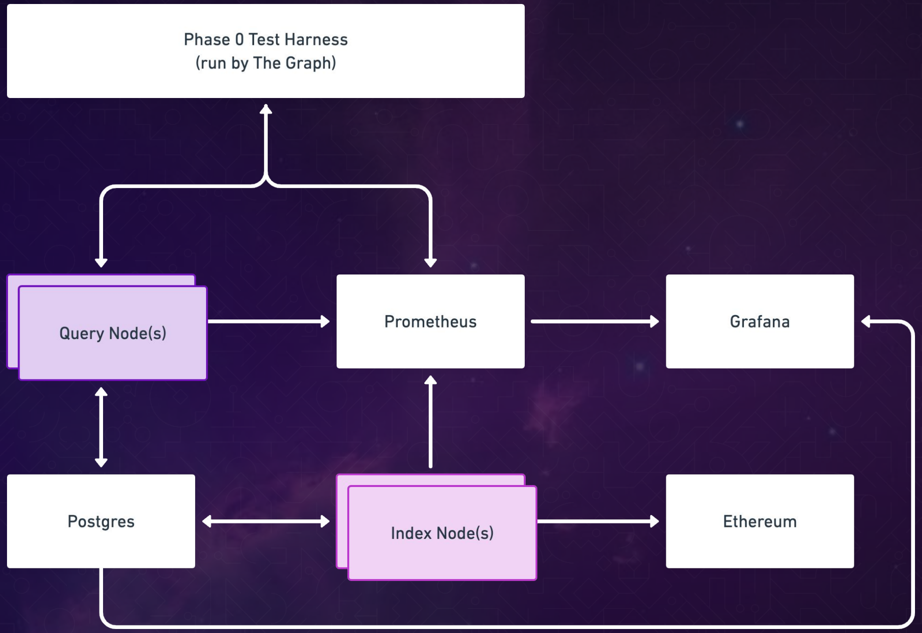
Infrastructure

Basic Infrastructure



Infrastructure

Recommended Infrastructure



Infrastructure

Terraform & Kubernetes Walkthrough

Kubernetes

- Containers
- Services & Ingress
- Grafana Dashboards
- GraphiQL Playground
- README
- Kustomize: Base vs. Custom

Terraform

- README

Infrastructure

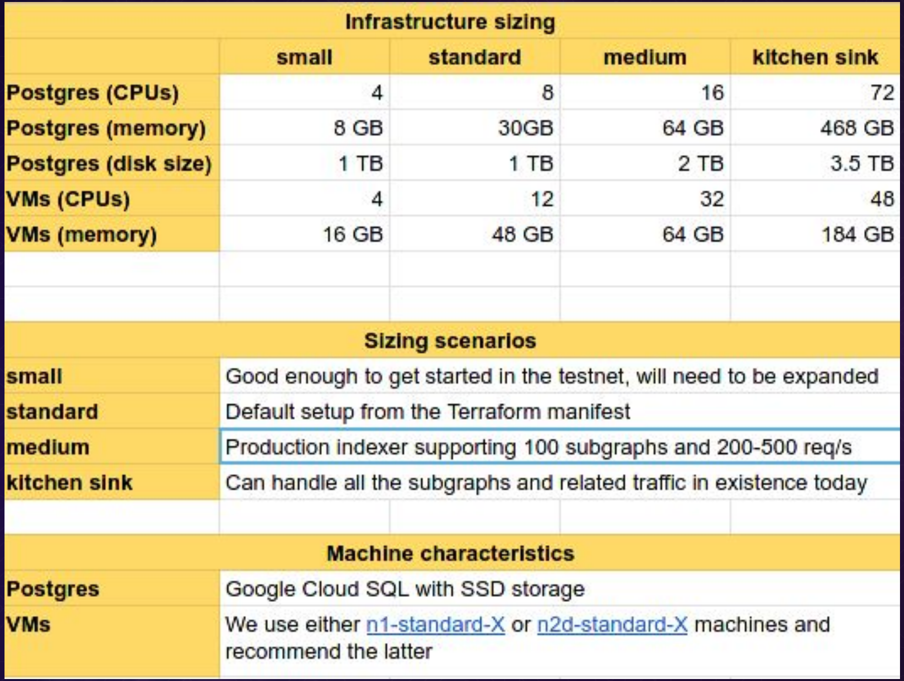
Ethereum

- Minimum required:
  Mainnet archive node
- Ideal: Mainnet archive node with OpenEthereum trace API
- Tested: OpenEthereum, Geth
- Own archive node (bonus points)
- But: Syncing may take too long for phase 0

Resource Guidance

Resource Guidance

Graph Infrastructure



Infrastructure sizing

	small	standard	medium	kitchen sink
Postgres (CPUs)	4	8	16	72
Postgres (memory)	8 GB	30GB	64 GB	468 GB
Postgres (disk size)	1 TB	1 TB	2 TB	3.5 TB
VMs (CPUs)	4	12	32	48
VMs (memory)	16 GB	48 GB	64 GB	184 GB

Sizing scenarios

small	Good enough to get started in the testnet, will need to be expanded
standard	Default setup from the Terraform manifest
medium	Production indexer supporting 100 subgraphs and 200-500 req/s
kitchen sink	Can handle all the subgraphs and related traffic in existence today

Machine characteristics

Postgres	Google Cloud SQL with SSD storage
VMs	We use either n1-standard-X or n2d-standard-X machines and recommend the latter

Resource Guidance

Ethereum

- Bandwidth requirements:
  Phase 0:
    - < 1M requests/day
    - < 20 GB/day
  Phase 1+:
    - ~ 3-4M requests/day
    - ~ 50-80 GB/day
- Load balancing

Monitoring

Monitoring

Logging

- Structured logging for queries and indexing
- Can easily be searched for subgraph deployments, block numbers etc. (demo)
- Supports Elasticsearch (CLI args)

Monitoring

Recommended Metrics

- Queries by response code
  - 2xx, 5xx
- Query latency (for successful requests; avg, median, 95th percentile)
- CPU load for query nodes, index nodes and database
- Memory usage of query nodes and index nodes
- Query node cache hits*

Monitoring

Query Logs

```
Jul 30 10:00:00.072 INFO
Query timing (GraphQL),
        complexity: <number>,
        block: <number>,
        cached: <hit|miss|shared|insert>,
        query_time_ms: <number>,
        variables: <json>,
        query: <GraphQL query>,
        subgraph_id: <subgraph id>,
        component: GraphQlRunner
```

Configuration

Configuration

Configuration mechanisms

- Terraform / Kubernetes
- Environment variables
- CLI args (custom Docker)
- Ethereum config file (auth)

Configuration

Environment Variables

- Ethereum tweaks
- Subgraph mappings limits
- GraphQL limits
- Query caching
- Misc
  - Node ID
  - Postgres connection pool size
  - GraphQL query logging
  - Load management

Configuration

Index Node Scaling

- Only one block ingestor
- Every index node must have a unique node ID
  - Included in Kubernetes setup (~ pod name)

Q&A