Running a local development environment

L2에서 스마트 컨트랙트를 개발하기 위해서는 L2 개발 환경을 만들어야 합니다. L1은 Ganache, hardhat 등을 활용하여 간단히 개발 환경을 만들 수 있는 반면에, L2를 위한 개발 환경은 프로젝트에 따라 더 많은 서비스를 실행해야 합니다. 본 섹션에서는 리눅스 기반 환경에서 Titan L2 개발 환경을 만드는 방법을 설명합니다.
Titan은 간단한 명령어 실행을 통한 L2 네트워크 구축을 지원합니다. 이를 통해 사용자들은 복잡한 설정 작업 없이 쉽게 네트워크를 관리할 수 있습니다. 더불어, Titan은 높은 확장성과 유연성을 가지고 있어 개발자들이 비즈니스 요구사항에 따라 네트워크를 조정할 수 있습니다. 이러한 이점들은 개발자들이 필요한 IT 인프라를 구축하는 데 큰 도움을 줄 것입니다.

Docker

Minimum Hardware Requirements

- Titan L2는 비교적 가벼운 스펙의 PC나 laptop에서 손쉽게 개발 환경을 구축하여 사용할 수 있습니다. 아래는 노드 실행을 위한 최소 하드웨어 사양입니다. (AWS EC2 t2.medium과 동일)

- CPU: 2 Core
- RAM: 4 GB

Prerequisite

- 본 가이드에서 OS는 Ubuntu 22.04 (LTS)로 특정하여 설명하겠습니다. Titan은 L2 개발 환경을 만들기 위해 Docker Engine를 사용합니다. Docker 컨테이너를 사용하면 애플리케이션의 배포, 확장 및 관리를 효과적으로 수행할 수 있습니다.
  - 설치 방법 참고 링크: https://docs.docker.com/engine/install/#server
- 위 설치 링크를 통해 최신 버전의 Docker Engine을 설치하면 `docker compose` 로 명령어 이름이 설정되어 있습니다. `docker-compose` 명령어를 사용할 수 있도록 아래와 같이 설정합니다.

```
echo docker compose '$*' | sudo tee /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

Titan monorepo

- git 유틸리티를 이용하여 Titan monorepo를 로컬 환경에 다운로드합니다.

```
git clone https://github.com/tokamak-network/tokamak-titan.git
cd tokamak-titan
```

Docker 빌드 및 Titan L2 네트워크 실행

- 로컬 개발 환경은 `ops` 디렉토리에 정의된 docker 컨테이너들을 활용합니다. 컨테이너로 만들어진 여러 패키지들을 통합해서 관리할 수 있고, 쉽게 실행하고 중단할 수 있습니다. 아래에서는 컨테이너들을 사용하는 방법을 설명하며, 모든 명령은 `ops` 디렉토리에서 실행해야 합니다.

디렉토리 변경

- Titan L2 네트워크 실행에 필요한 설정 파일인 `docker-compose.yml` 와 스크립트들이 위치한 `ops` 디렉토리로 이동합니다.

```
cd ops
```

빌드

- 사용자는 tokamak-titan 레포지토리 내의 패키지들을 직접 빌드하여 docker image를 생성할 수 있습니다. `docker-compose` 명령어 입력 시 입력하는 설정 파일의 이름이 `docker-compose.yml` 일 경우 생략할 수

있습니다

```
docker-compose -f ./docker-compose.yml build
```

- 그리고 빌드 없이 아래와 같이 사전에 배포된 docker image를 로컬에 다운로드할 수도 있습니다.

```
docker-compose -f ./docker-compose.yml pull
```

실행

- 로컬에 저장된 docker image를 실행하여 Titan L2 네트워크를 시작합니다.

```
docker-compose -f ./docker-compose.yml up -d
```

상태 확인

- 실행 중인 컨테이너의 상태를 확인합니다.

```
docker-compose -f ./docker-compose.yml ps
```

중단

- Titan L2 네트워크를 종료합니다.

```
docker-compose -f ./docker-compose.yml down
```

테스트

- integration_tests 컨테이너를 실행하여 Titan L2 네트워크를 테스트할 수 있습니다.

```
docker-compose run integration_tests
```

Simple Cluster: Titan L2 based on k8s

Titan은 k8s 기반 클러스터를 활용한 L2 인프라도 함께 제공하고 있습니다. Simple Cluster에서는 L2 개발 환경 구축을 위한 기본 구성요소 뿐만 아니라 블록/트랜잭션을 실시간으로 확인할 수 있는 Block Explorer와 L1-L2 간 자산 브릿징에 사용할 수 있는 Gateway Apps도 같이 포함되어 있어 빌더의 어플리케이션 테스트에 매우 유용합니다.

- 기본 구성요소: L1, L2, deployer, DTL (data transport layer), BSS (batch submitter service), relayer
- Apps: Block Explorer, Gateway

사용자는 쉘 스크립트를 이용하여 간편하게 Titan L2를 실행 및 중지할 수 있고, L1/L2/Explorer/Gateway에 접속하기 위한 퍼블릭 도메인을 제공받아 외부 네트워크에서도 자신의 L2에 접속하여 트랜잭션을 요청하고 네트워크를 확인할 수 있습니다.
k8s에서는 YAML이나 JSON 형식의 manifest 파일을 통해 k8s 오브젝트의 상태와 설정을 관리합니다. Simple Cluster에서 L2 기본 구성요소의 설정값은 configMap.yaml 파일 (예시. DTL의 configMap.yaml)에서 확인할 수 있습니다. 사용자는 L1, L2의 RPC 엔드포인트, BSS의 롤업 주기, confirmation 횟수 등 설정값을 자유롭게 변경하여 L2를 간편하게 커스터마이징할 수 있습니다.

Simple Cluster에서는 두 가지 가이드를 제공하고 있습니다.

- Guide 1: Titan 네트워크 실행 및 중지, Titan Apps 실행 및 중지, 퍼블릭 도메인 제공 (이미 리눅스 기반 서버를 가지고 있을 때. 서버는 Ubuntu 권장)
  - https://github.com/tokamak-network/tokamak-titan-simple-cluster/blob/main/README.md
- Guide 2: Terraform을 이용한 EC2 프로비저닝 (AWS 계정을 가지고 있으며 EC2에 Titan L2를 구축하고 싶을 때)
  - https://github.com/tokamak-network/tokamak-titan-simple-cluster/blob/main/terraform/README.md

Simple Cluster 레포지토리 주소와 L2 네트워크 구조를 확인할 수 있는 다이어그램은 아래 링크를 참고하세요.

- Github: https://github.com/tokamak-network/tokamak-titan-simple-cluster
- Diagram: https://github.com/tokamak-network/tokamak-titan-simple-cluster/blob/main/assets/k8s-titan-diagram.png