

Running a local development environment

⋮

To develop smart contracts on Titan L2, it is necessary to create an L2 development environment. While setting the development environment for L1 (Ethereum) is relatively simple, using tools such as Ganache or hardhat-node, L2 development environment requires additional resources. This section explains how to create a Titan L2 development environment in a Linux-based environment.

Titan supports L2 network deployment through simple command execution. This allows users to easily manage the network without complicated setup tasks. In addition, Titan is highly scalable and flexible, allowing developers to adjust the network according to business requirements. These benefits will go a long way in helping developers build the IT infrastructure they need.

Docker

Minimum Hardware Requirements

- Titan L2 can be easily deployed and used as a development environment on a relatively low-spec PC or laptop. Below are the minimum hardware specifications for running a node (same as AWS EC2 [t2.medium](#) spec)
 - CPU: 2 Core
 - RAM: 4 GB

Prerequisite

- The Operating system (OS) is assumed to be Ubuntu 22.04 (LTS) in this guide. Titan uses the Docker Engine to create an L2 development environment. Docker's containers allow you to effectively deploy, scale, and manage your applications.
 - installation: <https://docs.docker.com/engine/install/#server>
- When you installed the latest version of Docker Engine via the above installation link, the command name was set to `docker compose`. Make the following settings to enable the `docker-compose` command.

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

Titan monorepo

- Download Titan monorepo to your local environment using Git.

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

Docker Build & Run Titan L2 node

- The local development environment utilizes docker containers defined in the `ops` directory. Docker containers help consolidate and manage multiple packages and can be easily started and stopped. The following instructions outline how to use containers for Titan L2.

All commands must be run from the `ops` directory.

Change directory

- You navigate to the `ops` directory where `docker-compose.yml`, the configuration file required to run the Titan L2 network and the scripts.

cd ons

Build

- Users can create a docker image by building the packages in the [tokamak-titan](#) repository themselves. The configuration file you put when entering the `docker-compose` command can be omitted if it is named `docker-compose.yml`.

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

- You can also download a docker image without build packages, as shown below.

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

Run

- You can run the docker image to start the Titan L2 network.

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

Check Status

- You can check the status of a running container.

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

Stop

- You can shut down the Titan L2 network.

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

Test

- You can run the `integration_tests` container to test your Titan L2 network.

```
docker-compose run integration_tests
```

Simple Cluster: Titan L2 based on k8s

Titan also offers L2 infrastructure based on k8s. Simple cluster is not only the basic components for building an L2 development environment but also Apps which include Block Explorer and Gateway. The Block Explorer allows you to view blocks/transactions in real-time and the Gateway is used for bridging assets between L1 and L2. It is very useful for builders to test their applications.

- Basic Components: L1, L2, deployer, DTL (data transport layer), BSS (batch submitter service), relay
- Apps: Block Explorer, Gateway

The user can easily start and stop the L2 using shell scripts and check and access their L2 using a public domain of L1, L2, Block Explorer, and Gateway.

In k8s, we manage the state and settings for k8s objects through manifest files in YAML or JSON format. In Simple Cluster, the settings of the L2 basic components can be found in the `configMap.yaml` file (e.g., [configMap.yaml in DTL](#)). The user can simply customize L2 by changing the setting values like RPC endpoint, rollup interval of BSS, block confirmation, etc.

Simple Cluster provides two guides to help you get started.

- Guide 1: Start and stop the L2 network, start and stop Titan Apps, and provide a public domain (If you have a Linux-based server. we recommend Ubuntu OS)
 - <https://github.com/tokamak-network/tokamak-titan-simple-cluster/blob/main/README.md>
- Guide 2: Make EC2 with Terraform (If you have an AWS account and want to deploy L2 on EC2)

- <https://github.com/tokamak-network/tokamak-titan-simple-cluster/blob/main/terraform/README.md>

See the link below for a repository and network structure.

- 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>