Overview

Airdrop contract, used for TGE event, is just a [cw-tokens/cw20-merkle-airdrop-contract](#) with several patches:

- removed native token distribution functionality, since contract will only distribute cNTRN tokens, which are CW20 like tokens with 1:1 vested uNTRN;
- removedburn
- ,burn_all
- andwithdraw
- ExecuteMsg's;
- claim
- patched to issueAddVesting
- message to[credits contract](#)
- ;
- withdraw_all
- patched to burn cNTRN tokens and send (received in exchange for burning) NTRN tokens to[reserve contract](#)
- .withdraw_all
- can only be called after vesting period is over;
- stages logic removed, since we will only need one airdrop stage, also merged register merkle tree and instantiate messages into one;
- removed owner freeze logic;
- removed cross-chain airdrops;
- tests moved to separate file;
- enforced start/expiration logic to accept only timestamps, since block heights are not compatible with vesting logic.

Flow

First, a [merkle-airdrop-cli](#) tool is used to generate merkle root for all airdrop addresses and their corresponding airdrop amounts.

Second, an airdrop contract is deployed with credits address in genesis.

Third, user sends claim message to airdrop contract and specifies their amount and merkle proof. If merkle proof can be validated successfully according to merkle root specified in instantiate message in genesis, user then receives their cNTRN tokens, and a corresponding vesting period is set for them in[credits contract](#) .