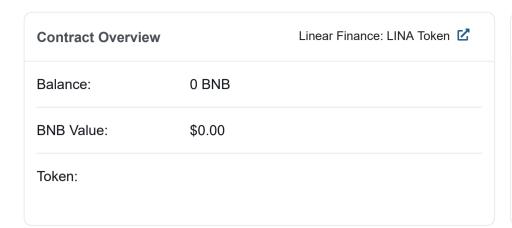
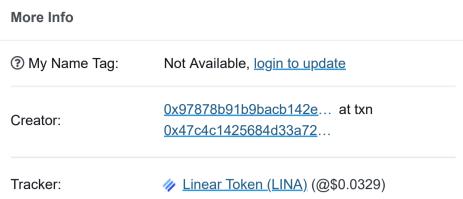
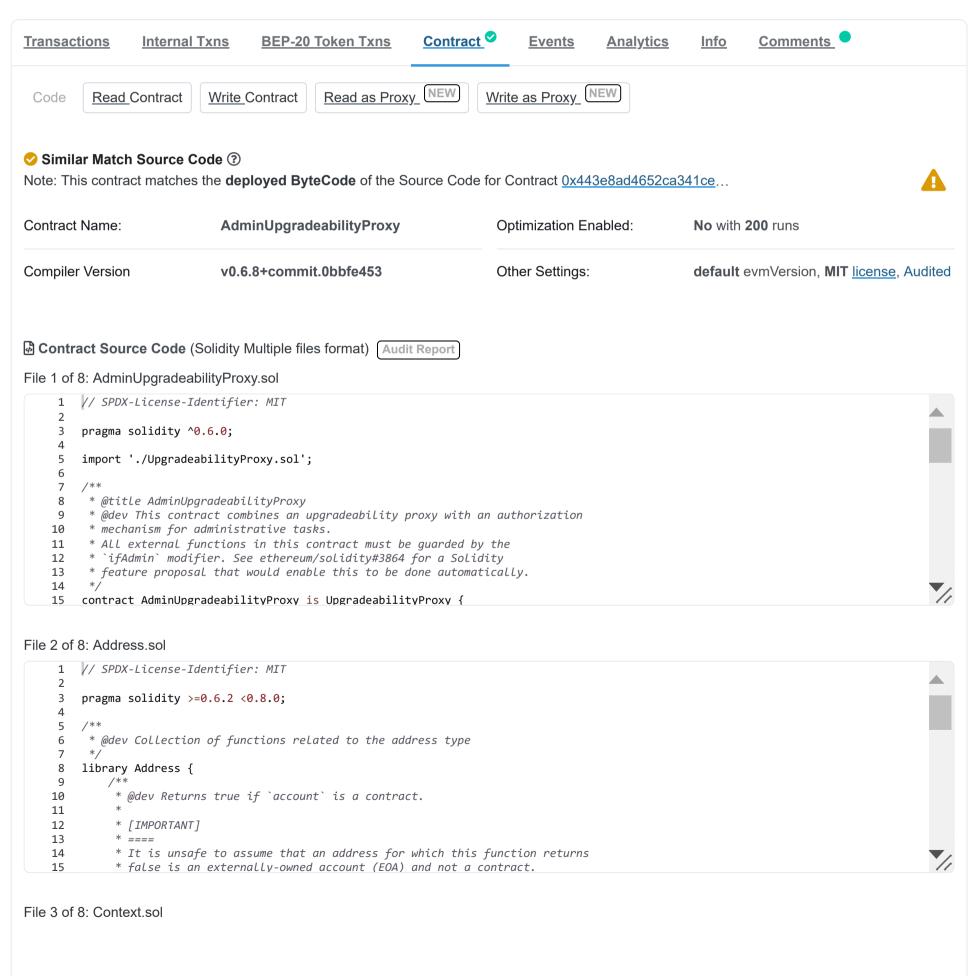
Token Contract

Sponsored: // - **AAX** - Sign to earn 50 USDT rewards! Visit **AAX.com** now!









```
// SPDX-License-Identifier: MIT
    1
       pragma solidity >=0.6.0 <0.8.0;</pre>
    3
    4
    5
        * @dev Provides information about the current execution context, including the
    6
        * sender of the transaction and its data. While these are generally available
        * via msg.sender and msg.data, they should not be accessed in such a direct
         * manner, since when dealing with GSN meta-transactions the account sending and
         * paying for execution may not be the actual sender (as far as an application
   10
        * is concerned).
   12
        * This contract is only required for intermediate, library-like contracts.
   13
   14
   15 abstract contract Context {
File 4 of 8: Initializable.sol
    1 // SPDX-License-Identifier: MIT
       // solhint-disable-next-line compiler-version
       pragma solidity >=0.4.24 <0.8.0;
    6
    7
       /**
        * @dev This is a base contract to aid in writing upgradeable contracts, or any kind of contract that will be deployed
    8
        * behind a proxy. Since a proxied contract can't have a constructor, it's common to move constructor logic to an
        * external initializer function, usually called `initialize`. It then becomes necessary to protect this initializer
   10
   11
         * function so it can only be called once. The {initializer} modifier provided by this contract will have this effect.
   12
        * TIP: To avoid leaving the proxy in an uninitialized state, the initializer function should be called as early as
         * possible by providing the encoded function call as the `_data` argument to {UpgradeableProxy-constructor}.
   14
   15
File 5 of 8: Ownable.sol
       // SPDX-License-Identifier: MIT
    3
       pragma solidity >=0.6.0 <0.8.0;</pre>
    4
       import "./Context.sol";
    5
    7
        * @dev Contract module which provides a basic access control mechanism, where
        * there is an account (an owner) that can be granted exclusive access to
         * specific functions.
    9
   10
        * By default, the owner account will be the one that deploys the contract. This
   11
         * can later be changed with {transferOwnership}.
   12
   13
        * This module is used through inheritance. It will make available the modifier
        * `onlyOwner`, which can be applied to your functions to restrict their use to
File 6 of 8: Proxy.sol
         · IL WEJ LITES A JULIUDUCK JUTICLION CHAL WELEGULES ALL CALLS TO THE WARTESS
   10
        * returned by the abstract _implementation() internal function.
   11
        abstract contract Proxy {
   13
          * @dev Fallback function.
   14
           * Implemented entirely in `_fallback`.
   15
   16
          fallback () payable external {
   17
           _fallback();
   18
   19
   20
   21
   22
           * @dev Receive function.
           * Implemented entirely in `_fallback`.
   23
File 7 of 8: ProxyAdmin.sol
       // SPDX-License-Identifier: MIT
    1
    2
    3
       pragma solidity ^0.6.0;
      import "./Ownable.sol";
    6 import "./AdminUpgradeabilityProxy.sol";
    8
        * @title ProxyAdmin
        * @dev This contract is the admin of a proxy, and is in charge
   10
         * of upgrading it as well as transferring it to another admin.
   11
   12
       contract ProxyAdmin is Ownable {
   13
   14
   15
```

File 8 of 8: UpgradeabilityProxy.sol

```
// SPDX-License-Identifier: MIT
 3
    pragma solidity ^0.6.0;
    import './Proxy.sol';
    import './Address.sol';
 8
     * @title UpgradeabilityProxy
     st @dev This contract implements a proxy that allows to change the
10
11
     * implementation address to which it will delegate.
     * Such a change is called an implementation upgrade.
12
13
14
    contract UpgradeabilityProxy is Proxy {
15
```

Contract Security Audit

• Callisto Network- Dec 30th, 2021 - Security Audit Report

≅ Contract ABI

```
[{"inputs":[{"internalType":"address","name":"_logic","type":"address"},
{"internalType":"address","name":"_admin","type":"address"},
{"internalType":"bytes","name":"_data","type":"bytes"}],"stateMutability":"payable","type":"constructor"},
{"anonymous":false,"inputs":[{"indexed":false,"internalType":"address","name":"previousAdmin","type":"address"},
{"indexed":false,"internalType":"address","name":"newAdmin","type":"address"}],"name":"AdminChanged","type":"event"},
{"anonymous":false,"inputs":
[{"indexed":true,"internalType":"address","name":"implementation","type":"address"}],"name":"Upgraded","type":"event"},
{"stateMutability":"payable","type":"fallback"},{"inputs":[],"name":"admin","outputs":
[{"internalType":"address","name":"","type":"address"}],"stateMutability":"nonpayable","type":"function"},{"inputs":
[{"internalType":"address","name":"newAdmin","type":"address"}],"name":"changeAdmin","outputs":
```

</> Contract Creation Code

Ⅲ Deployed ByteCode Sourcemap



Swarm Source

ipfs://b0503c23d64d00872f309af6ebc439401f4a3a5289bbb698fa4e5aabd7476426