
DZap NFT Staking Task

Introduction

The DZap NFT staking system utilizes two distinct contracts: the **NFT Staking Contract** and the **RewardToken Contract**. The system leverages the Universal Upgradeable Proxy Standard (UUPS) to ensure upgradability, security, and flexibility.

Contract 1: NFT Staking Contract

Purpose

The **NFT Staking Contract** manages the staking and unstaking of ERC721 NFTs. It uses the UUPS proxy pattern to support future upgrades while retaining the state and avoiding redeployment.

Logic Explanation

Staking Management

- **Functionality**: Users can stake ERC721 NFTs by transferring them to this contract.
- **Implementation**: The `stake` function ensures the sender is the owner of the NFT, transfers the NFT to the contract, and records the staking details in a mapping.
- **Reason**: This design guarantees that only the authorized NFT owner can stake, and it maintains accurate records for reward calculations.

Unstaking Process

- **Functionality**: Users can unstake their NFTs after a defined unbonding period.
- **Implementation**: The `unstake` function checks that the unbonding period has elapsed, transfers the NFT back to the user, and updates the staking records.
- **Reason**: This prevents users from unstaking NFTs before the completion of the required period and ensures proper return of NFTs.

Claiming Rewards

- **Functionality**: Users can claim rewards based on the duration their NFTs have been staked.

- **Implementation**: The `claimRewards` function calculates the total reward, verifies that the claim delay has passed, and transfers the rewards. It updates the last claim timestamp.
- **Reason**: This ensures rewards are distributed proportionately to the staking duration and prevents premature claims.

Stake Tracking

- **Functionality**: Keeps track of all NFTs staked by each user and their respective staking details.
- **Implementation**: Utilizes mappings and arrays to efficiently manage staking data.
- **Reason**: This structure facilitates the management of staked NFTs and accurate reward calculations.

Internal Functions

- **Functionality**: Includes helper functions for reward calculation and NFT management.
- **Implementation**: Functions like `_calculateRewards` and `_removeNFT` aid in reward computation and NFT management.
- **Reason**: Ensures accurate reward distribution and efficient handling of NFTs.

Events

- **Functionality**: Emits events for staking, unstaking, and reward claims.
- **Implementation**: Events such as `NFTStaked`, `NFTUnstaked`, and `RewardsClaimed` notify the network about these actions.
- **Reason**: Provides transparency and allows external systems to track contract activities.

Upgradability with UUPS Proxy

- **Functionality**: Employs the UUPS proxy pattern for contract upgrades.
- **Implementation**: Inherits from `UUPSUpgradeable` and overrides `_authorizeUpgrade` to restrict upgrade permissions.
- **Reason**: This allows future upgrades without losing contract state or requiring redeployment, enhancing adaptability.

Pause and Resume Staking

- **Functionality**: Allows the contract owner to pause and resume staking operations.
- **Implementation**: Functions `pauseStaking` and `resumeStaking` provide administrative control.
- **Reason**: Offers flexibility for maintenance or emergency situations.

Contract Code

The `StakeNFT` contract utilizes OpenZeppelin's `OwnableUpgradeable`, `UUPSUpgradeable`, and `ReentrancyGuardUpgradeable` to provide ownership control, upgradeability, and protection against reentrancy attacks.

Contract 2: RewardToken Contract

Purpose

The **RewardToken Contract** is an ERC20 token used to distribute rewards within the staking system. It is designed to operate independently from the staking logic but is essential for rewarding users.

Logic Explanation

Reward Token Initialization

- **Functionality**: The contract initializes with a fixed supply of tokens.
- **Implementation**: The `RewardToken` contract extends the ERC20 standard and mints an initial supply of tokens to the deployer.
- **Reason**: Provides a consistent and pre-defined reward currency for the staking system.

Contract Code

The `RewardToken` contract is a standard ERC20 implementation with an initial minting of 1,000,000 tokens to the contract deployer.

Summary

The combined use of the **NFT Staking Contract** and **RewardToken Contract** creates a robust staking system. The UUPS proxy pattern employed in both contracts ensures the system's long-term viability and adaptability, allowing for future enhancements without compromising existing functionalities.