

How Contributors Can Recover Their Bond from Contribute Alpha Staking Contracts

Summary

This document outlines one way that users of <https://contribute.olas.network> (Contributors) who staked OLAS to the below staking contracts may recover their bond and any additional funds earned:

Staking contracts:

- **Contribute Alpha 1:** [Link to Contract](#), potential funds available: 50¹ OLAS (representing 50% of the contributor's stake) plus an additional 30% premium per *contributor wallet*.
- **Contribute Alpha 2:** [Link to Contract](#), potential funds available: 50² OLAS (representing 50% of the contributor's stake) plus an additional 30% premium per *contributor wallet*.
- **Contribute Alpha 3:** [Link to Contract](#), potential funds available: 250³ OLAS (representing 50% of the contributor's stake) plus an additional 30% premium per *contributor wallet*.

Disclaimer for Proposal Execution

Please note that this process is only possible if the DAO approves the on-chain vote to fund the contributor bond deposits and premiums. The proposal can be tracked here: [DAO Proposals](#)⁴. If approved, the process can be executable starting from proposal execution time which, in turn, can happen from February 9 2025, 00:00 UTC.

Disclaimer for OLAS Sites

¹ Note that, before the end of the process you may also recover the other half of your stake.

² Note that, before the end of the process you may also recover the other half of your stake.

³ Note that, before the end of the process you may also recover the other half of your stake.

⁴To track the correct proposal [here](#), please refer to the one created on February 3, 2025, titled: *Fund Contribute Manager Recoverer contract to refund Contributors with a premium on Base via funds from Timelock Treasury. The initial Contribute implementation has a bug in the Contribute Manager contract which prevents full withdrawal of staked funds by stakers. All the funds (without the premium) are fully recoverable and can be returned back to Timelock Treasury with a subsequent on-chain vote. Additionally, remove nominees for staking contracts affected by the Contribute Manager bug. In accordance with Autonolas DAO Constitution at*
ipfs://bafybeibrhz6hnxsxcbv7dkzerq4chssotexb276pidzwclbytzj7m4t47u

No warranties are provided. Check [the disclaimer](#) for more details.

Disclaimer for this document

1. THIS DOCUMENT IS PROVIDED "AS IS" AND "AS AVAILABLE," AT YOUR OWN RISK, AND WITHOUT WARRANTIES OF ANY KIND. Neither Olas nor Valory nor any other party will be liable for any loss, whether such loss is direct, indirect, special or consequential, suffered by any party as a result of their use of this document.
2. None of the information available on this document, or made otherwise available to you in relation to its use, constitutes any legal, tax, financial or other advice. Where in doubt as to the action you should take, please consult your own legal, financial, tax or other professional advisors.
3. No one should use technologies, particularly emerging technologies, blockchains and related infrastructure like bridges, without fully understanding the risks involved.

One Way to Recover Your Bond and Premium

Users may use the exact order described below:

Step 1: Connect to the ContributeManager Contract

1. Connect your *contributor wallet* (the wallet you used to sign up for [Contribute](#), that owns the staked serviceID) to [ContributeManager Contract on BaseScan](#) by pressing the "Connect to Web3" button.



Step 2: Get your service Id and multisig address

2. You can recover your information as follows:
 - a. If you are still staked, you can recover your information going to the [ContributorsProxy](#) contract and query the function [mapAccountServiceInfo](#) with your *contributor wallet* address as an input parameter.

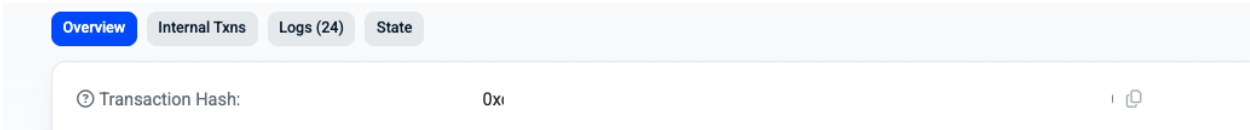
4. mapAccountServiceInfo (0x8b494a39)

Query

socialId uint256, serviceId uint256, multisig address, stakingInstance address

Take note of the second and third output variables *serviceId* and *multisig* address

- b. If you are not, you may proceed as follows:
- i. Search in the list of transactions signed with your *contributor wallet*, the one you signed to stake and search it on base scan. You should see something similar to the following image.



- ii. Click the button “Logs”



- iii. Search for *CreatedAndStaked*. You should see something similar to the following image. Note that, near serviceOwner you should see the address of your *contributor wallet*.

208

Address

0xaea9ef993d8a1a164397642648df43f053d43d85

CreatedAndStaked

(index_topic_1 uint256 socialId, index_topic_2 address serviceOwner, uint256 serviceId, index_topic_3 address multisig, address stakingInstance)

View Source

Topics

0

0xf1c69b12e526d2d4145f47d216fa261c6f468d177601c9081a0bcd86a4fa744f

1: socialId

Dec

⇒

2: serviceOwner

Dec

⇒

0x...

3: multisig

Dec

⇒

0x...

Data

serviceId: 176

stakingInstance: 0x95146Adf659f455f300D7521B3b62A3b6c4aBA1F

At this point, you can take note of *serviceId* and *multisig* address displayed there.

Step 3: Unstake

3. If not unstaked previously, once connected to the ContributeManager contract, click the “Write” button under the [unstake method](#) to initiate the unstaking process.

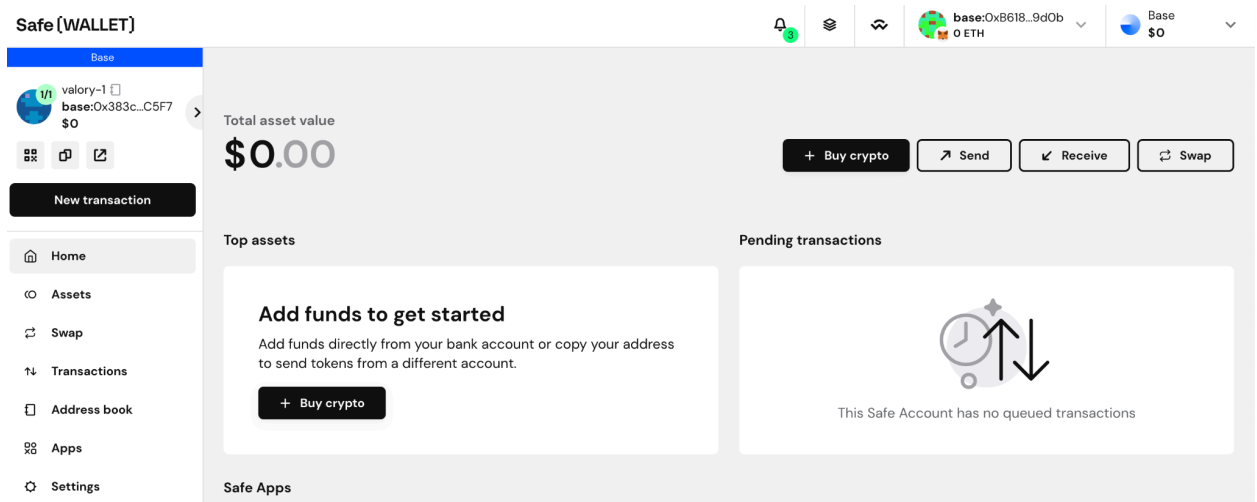


Step 4: Slash

5. Visit the Safe App website and connect your *contributor wallet* by using the link below:

<https://app.safe.global/home?safe=base:multisig>

One can replace the “multisig” portion of the URL with the multisig address you obtained in the previous step. Once done, the page should appear as shown below:



Note: The image above is for illustrative purposes only. Please use your own data as described above.

6. Once connected to the Safe App, the [slash method](#) can be executed by following the steps described below:
 - Click “New transaction” button;

New transaction

- Click “Transaction Builder” button

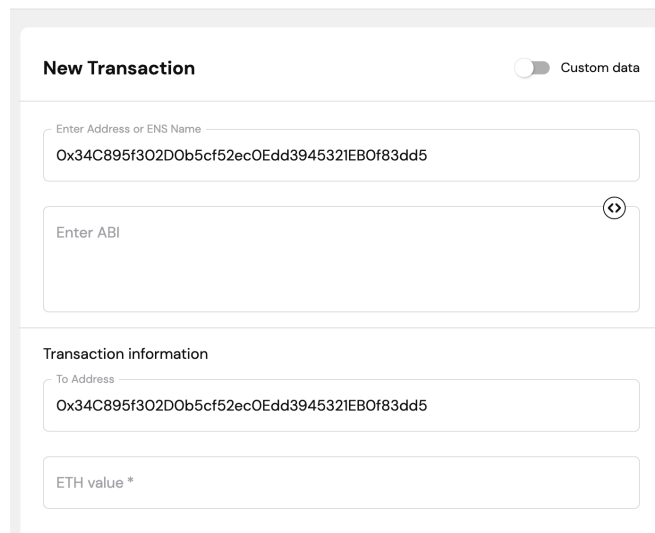
Interact with contracts

 Transaction Builder

- Once the Transaction build window opens up, include the following address in “Enter address” field: [0x34C895f302D0b5cf52ec0Edd3945321EB0f83dd5](#)
 - At this point, the ABI usually loads automatically. If this is the case, choose in “Contract method selector” the *slash* method
 - If the ABI field continues staying empty as in the following image. Insert the following text in the “Enter ABI” field:

```
[{"inputs":[{"internalType":"address[]","name":"agentInstances","type":"address[]"}, {"internalType":"uint256[]","name":"amounts","type":"uint256[]"}, {"internalType":"uint256","name":"serviceId","type":"uint256"}], "name":"slash", "outputs":[{"internalType":"bool","name":"success","type":"bool"}], "stateMutability":"nonpayable","type":"function"}]
```

Transaction Builder ⓘ



The screenshot shows the 'Transaction Builder' interface. At the top, there's a 'New Transaction' header with a 'Custom data' toggle. Below this, there's a section for 'Enter Address or ENS Name' with the address '0x34C895f302D0b5cf52ec0Edd3945321EB0f83dd5' entered. Below that is the 'Enter ABI' field, which is empty. At the bottom, there's a 'Transaction information' section with a 'To Address' field containing the same address and an 'ETH value *' field.

Ultimately, the form with the chosen slash function should appear as in the following figure:

Transaction information

To Address

Ox34C895f302D0b5cf52ec0Edd3945321EB0f83dd5

Contract Method Selector

slash

agentInstances (address[]) *

amounts (uint256[]) *

serviceld (uint256) *

+ Add new transaction

- Populate this form using the following parameters:
 - **agentInstances** = [address of the contributor staker (the one used to connect in safe app)]
 - **amounts** = [bond deposit in wei]:
 1. 5000000000000000000 for Contribute Alpha 1 & 2
 2. 2500000000000000000 for Contribute Alpha 3
 - **serviceld** = *serviceld* recovered in step 3.

Note: The image below is for illustrative purposes only. Please use your own data as described above.

Transaction information

To Address

Ox34C895f302D0b5cf52ec0Edd3945321EB0f83dd5

Contract Method Selector

slash

agentInstances (address[]) *

[0xB618970ff99562D0D27b756256b7da55A16b9d0b]

amounts (uint256[]) *

[5000000000000000000]

serviceld (uint256) *

179

+ Add new transaction

- Click “Add new transaction” button at the bottom of the same screen (see previous image);
- Click “Create Batch” button;

1 Transactions Batch

1 base:0x34C8...3dd5 slash

Create Batch

- Click “Send Batch” button;

Review and Confirm

1 Transactions Batch

1 base:0x34C8...3dd5 slash

Send Batch Cancel Simulate

- Leave everything as is in the next form and click the “Execute” button, then sign with your *contributor wallet* when prompted.

Execute
You're about to create and execute this transaction.

Would you like to execute the transaction immediately?

☒ Yes, **execute** ☐ No, later

Proceed with caution
The transaction could not be checked for security alerts. Verify the details and addresses before proceeding.

Estimated fee ⚠ Cannot Estimate

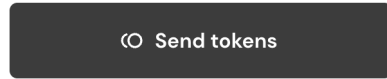
Who will pay gas fees:
☒ Sponsored by Safe ☐ Connected wallet

5 free transactions left today ⓘ

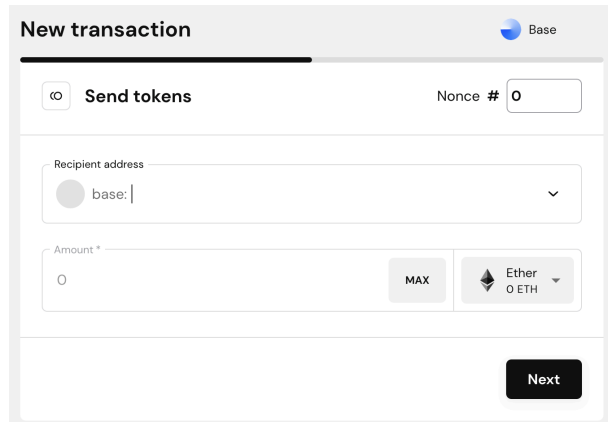
6. If you happened to have OLAS token rewards for staking, those were transferred to your multisig during the unstaking process. You can transfer them to any chosen address by doing the following from the Safe App:
 - a. Click “New transaction” the same way as at the beginning of the previous step;

- b. Click “Send tokens” button;

Manage assets



- c. Input a recipient address on Base, select OLAS token, max amount, and click “Next”;

A screenshot of a web interface titled "New transaction" in the top left corner. The interface is for the "Base" network, indicated by a logo in the top right. Below the title bar, there is a tab labeled "Send tokens" with a circular icon. To the right of this tab is a "Nonce # 0" input field. Below the tab, there is a "Recipient address" input field with a dropdown menu showing "base:". Below that is an "Amount *" input field with a "MAX" button and a token selector showing "Ether 0 ETH". At the bottom right of the form is a dark gray "Next" button.

- d. Sign and execute to finalize token transfer in a similar way with the previous step.

Step 5: Connect to the ServiceManagerToken Contract

7. Connect your *contributor wallet* to [ServiceManagerToken Contract on BaseScan](#) by pressing the “Connect to Web3” button.



Step 6: Terminate

8. Once connected to the ServiceManagerToken contract, use the [terminate method](#) method with the following input parameter:
- o **serviceld** = *serviceld* recovered in step 3.

Note that, you may have received half of the stake in your *contributor wallet* at the end of this point.

Note: The image below is for illustrative purposes only. Please use your own data as described above.

10. terminate (0x7a828b28)

serviceId (uint256) +

179

Write

Step 7: Connect to the RecovererContributeManager Contract and recover the bond

9. Connect your *contributor wallet* to [RecovererContributeManager Contract on BaseScan](#) by pressing the “Connect to Web3” button.

Code Read Contract Write Contract

Connect to Web3

[Expand all] [Reset]

10. Once connected to the RecovererContributeManager contract, execute the [recover method](#) with the following input parameters:
 - a. **serviceId** = *serviceId* recovered in step 3.

Note: The image below is for illustrative purposes only. Please use your own data as described above.

3. recover (0x0ca35682)

The service must be unstaked from ContributorManager and terminated.
Recovers bond amount for contributors.

serviceId (uint256) +

179

Service Id.

Write

At this point, if eligible users have followed these steps successfully, they may have their stake deposit credited to their *contributor wallets*, along with the additional premium offered by the DAO.
