

## Logris Vaults (Alchemix Leveraged Vaults)

The most common question I've been asked regarding my [Double Logris](#) post is how to do it. The unfortunate answer at the time was that it wasn't possible because:

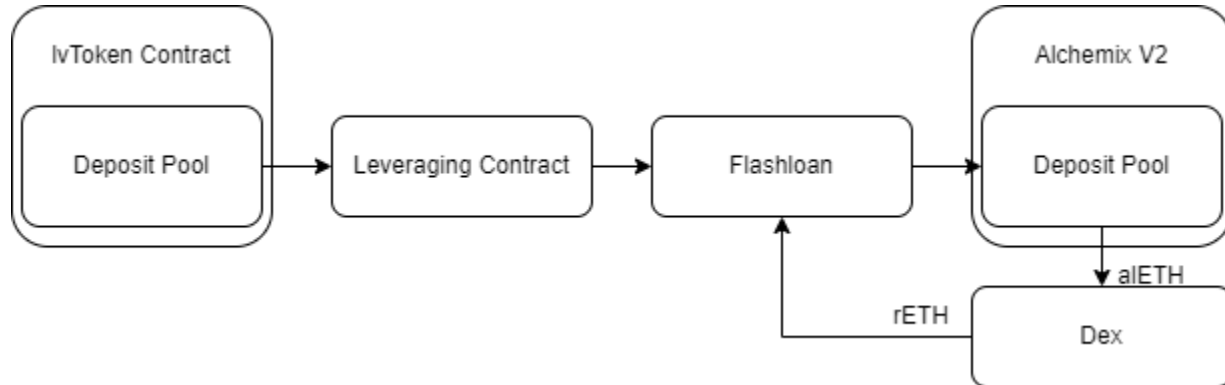
- 1) Smart contract calls were disabled and were necessary for the flash loan functionality.
- 2) The collateralization ratio for aETH was 400%, not 200% so while the concept was sound it wasn't going to double your interest.

With the launch of Alchemix v2 and the change of the collateralization ratio for rETH and stETH it is now possible for whitelisted contracts to get a 2x base yield. Unfortunately, no one has been whitelisted yet. That leaves open a collaborative opportunity.

Alchemix Leveraged Vaults are a system like Yearn vaults that implement a leveraged yield strategy using Alchemix as the source of yield. Just like Yearn vaults, to the user this is as simple as depositing the token corresponding to the Alchemix vault to receive an lvToken LP that just appreciates in value like cTokens do while using maximum Alchemix leverage as the yield strategy. The lvTokens are a [ERC-4626 token](#) which basically just means they appreciate in redemption value. Officially the lv prefix stands for leveraged vault. Internally I call these Logris Vaults.

This is a contract with some pros and cons compared to a Double Logris transaction. For one thing, the funds are escrowed. The Alchemix debt doesn't appear on your address. From a tax perspective, this is simpler. Toss money in, hold onto the LP, sell later and pay capital gains. However, it loses the capital loss offset that an unwrapped leveraged position would get. Because assets are pooled it gets the same gas efficiency benefits of your standard yield aggregators. While Defisaver or Furucombo could get whitelisted to enable a Double Logris transaction it would still be complicated to do it correctly. Most people have no experience taking a flash loan and would have trouble calculating how large a loan to take while accounting for loan fees and slippage on the aETH->ETH trade. So it's more convenient, gas efficient, and simpler for accounting but loses some of the tax loophole aspect that originally excited me.

## Component Overview



Each subsection below refers to a component in the diagram above.

### IvToken Contract

Each yield token has a unique wrapping Iv [ERC-4626 token](#). This contract includes a deposit pool of funds that have been deposited to escrow but not yet deposited in bulk to Alchemix through the leveraging contract.

This is also the contract with all the public functions for deposit/withdraw, admin only functions to manage internal variables and emergency shutdown, and whitelisted functions that releverage the vault to maintain the maximum APR.

### Leveraging Contract

Maxes out the leverage for the yield token using assets in the deposit pool and any outstanding credit in the Alchemix v2 contract. Here's a basic sequence of how it works:

- 1) Calculate available yield tokens (e.g. rETH) using deposit pool assets and withdrawable assets from Alchemix v2 (forgiven debt/interest).
- 2) Calculate amount needed for flash loan assuming maximum slippage on aToken trade.
- 3) Aggregate a lump sum deposit. This may include flash loaning yield assets.
- 4) Deposit the lump sum to Alchemix v2.
- 5) Borrow as much as required to repay the flash loan after accounting for slippage.
- 6) Trade for flash loaned assets.
- 7) Repay flash loan.

This entire process needs to scale down to the debt limit if it would be hit by the borrow step. FooBar (Alchemix Dev) has a starting point for this work in his v1->v2 migration tool.

### [Alchemist V2 Contract](#)

The main public contract for Alchemix V2 containing the deposit, mint, liquidate, and withdraw functions.

### Flashloan

This is a system such as Aave from which we can borrow the yield bearing asset. If none exists we may have to borrow a base asset like ETH and swap it for the yield bearing asset as part of the leveraging process.

### Dex

Your standard token trading mechanism. Probably going to use Balancer for rETH since that's where most of the liquidity is. stETH is probably going to use Curve.

## Relationship with AlchemixDAO

I talked to the Alchemix Dev's about this idea last year and they agreed the basic design is sound but I was told they were going to launch a grants program, I could be a guinea pig, and I should hold off. That program never materialized or it turned into a hackathon instead. I'm hoping to BUIDL this at EthDenver instead. The plan is for the Alchemix DAO to assume ownership of the contracts once developed. The primary benefit is this allows the Alchemix DAO to integrate the releveraging call with their AMO so they can minimize slippage. It *may* be possible to integrate the releveraging call with the call that raises the debt ceiling so these vaults get dibs on new capacity. This would turn what is otherwise a gas race into a system that finds an APR below 2x capacity when the debt ceilings are throttling adoption.

## Limited Scope

MVP:

- 1) Solidity Contracts of the Alchemix Leveraged Vaults for rETH and stETH. These should be reasonably extensible to other yield tokens without refactoring.
- 2) Comprehensive tests of the contracts.
- 3) A workflow to setup a local environment and run the tests.
- 4) Documentation/specs for contracts comparable to the Alchemix v2 Gitbook.

Time permitting we may build:

- 1) A proof of concept frontend.
- 2) Alchemix Leveraged Vault contracts for one or more stablecoins.

We definitely will not build:

- 1) A hosted environment for documentation or a testnet.
- 2) Any support for non-EVM compatible chains.