

Multicall operations

Now you've mastered execute function! Well, almost.

There's one more thing.

It's the heart of velcore, operation chaining, which will make your work more efficient!

Let's compound interest at VC-ETH pool. You should chain these 3 operations.

1. Harvest LP reward (VC emission)
2. swap VC to VC-ETH LP
3. Stake
- 4.
5. It's important to note that there is no separate process to swap the halving of VC for ETH. LPs always accept imbalanced deposits, so after receiving VC, they return the exact amount of LPs.
6. If you want to create an ETH-USDC LP after claiming VC, you can additionally swap all VC -> USDC and then create an ETH-USDC LP with that USDC.
7. By skipping the intermediate step, you can save a lot of gas and reduce the probability of revert.
8. In the process of chaining the OP, it is possible to use ALL
9. the balance of the intermediate steps or the EXACT
10. amount, but there is no
11. such command as using HALF, so it is good to consider this when creating LPs.
- 12.

Related tokens : VC, VC-ETH LP (We don't need to handle ETH in the intermediate step!)

Op type used :

SWAP(00) - buy LP

GAUGE(01) - Claim LP reward, stake

Interacting with : VC-ETH pool(=gauge for volatile pair)

So I should define 2 Tokens and 3 operations.

...

```
Copy // 2 tokens involved. index 0x00 : VC, index 0x01 : vc-eth lp
Token[] memory tokens = new Token[2];
tokens[0] = toToken(IERC20(vc));
tokens[1] = toToken(IERC20(vc_eth_lp));
```

```
VelcoreOperation[] memory ops = new VelcoreOperation[3]; // op[0]=Harvest VC : optype GAUGE / expect to 'receive'
VC(AT_MOST 0) // Gauge address is same with pool(lp) address only for volatile pools. addressvc_eth_gauge=vc_eth_pool;
ops[0].poolId = toPoolId(GAUGE,vc_eth_gauge);
ops[0].tokenInformations = new bytes32[2];
ops[0].tokenInformations[0] = toTokenInfo(0x00,AT_MOST,0);
ops[0].data = "";
```

```
// op[1]=Buy LP : optype SWAP / use all VC internal balance and expect to receive VC-ETH LP.
ops[1].poolId = toPoolId(SWAP,vc_eth_pool); // if amount type is ALL, you could just set 0 at the desiredAmount.
ops[1].tokenInformations = new bytes32[2];
ops[1].tokenInformations[0] = toTokenInfo(0x00,ALL,0);
ops[1].tokenInformations[1] = toTokenInfo(0x01,AT_MOST,0);
ops[1].data = "";
```

```
// op[2]=Stake LP : optype GAUGE / use ALL vc_eth_lp
ops[2].poolId = toPoolId(SWAP,vc_eth_pool); // Since we've harvested VC right before, we don't have any VC to claim. // but usually, we need to add tokeninfo to harvest VC when staking LP.
ops[2].tokenInformations = new bytes32[2];
ops[2].tokenInformations[0] = toTokenInfo(0x01,ALL,0);
ops[2].data = "";
```

```
return execute(tokens,newint128,ops);
```

...

Last updated 1 month ago On this page