

Deploying a CoW AMM Using Balancer

In this short tutorial, we describe how one can create and deploy their own Balancer CoW AMM.

You may also deploy a CoW AMM through the Balancer UI using [this guide](#).

The proposed transactions and interactions can be executed through any smart contract or EOA transaction builder, Etherscan "Write Contract," from other UI mechanisms, or directly from the console.

The current factory contract addresses are the following:

- Ethereum Mainnet: [0xf76c421bAb7df8548604E60deCCcE50477C10462](#)
- Gnosis Chain: [0x703Bd8115E6F21a37BB5Df97f78614ca72Ad7624](#)
- Sepolia Testnet: [0x1E3D76AC2BB67a2D7e8395d3A624b30AA9056DF9](#)

These contracts can be verified through [this](#) repository.

1. In order to create a new pool, one first needs to call the `newBPool`
2. function of the factory contract. An example transaction that creates a new pool can be found [here](#)
3. . The relevant `LOG_NEW_POOL`
4. log also reveals the address of the newly created pool; in our example transaction, it is `0x81530e9B069c69F6671A0A9d7Ee337cafEF419F6`
5. .

note In case ABI of the factory contract is not fetched, the ABI from the Sepolia network can be used. 1. The next step is to approve the tokens we want to add to the pool; this can be done by calling the `approve()` 2. function on the contract of the relevant ERC-20 token that is being added to the pool, where the "spender" needs to be set to the newly created BPool address from (1).

caution One needs to take the token's decimals into account in order to set the correct approval. 1. The next step is to bind the approved tokens to the pool, by using the `bind()` 2. function of the newly created pool contract. We need to do one bind per token that will be added to the pool. Note that the `denorm` 3. parameter should always be set to `1000000000000000000` (i.e., 10^{18}). 4. We can then set the swap fee; this is the fee traders will pay if they trade permissionlessly (i.e., outside of the batch), and this is done by calling the `setSwapFee()` 5. function of the pool contract. The fee parameter needs to be specified in wei, hence, a 10% fee would be `100000000000000000` (i.e., 10^{17})

note In order to guarantee full LVR protection, the fee should be set to 100%. caution The fee parameter is an unsigned integer where 10^{18} corresponds to 100% of fee. Note that 100% is actually not a valid value though, but one can set up to 99.99%. 1. The final step is to finalize the pool by calling the `finalize()` 2. function of the newly created pool contract. [Edit this page](#) [Previous Tutorials](#) [Next CoW AMM Liquidity for Solvers](#)