

Create your xERC20 Token Bridge

This guide walks you through deploying an xERC20 Warp Route between two networks of your choice. We'll use Ethereum as an example for the origin chain; however, you can select any supported network as the origin and destination.

Prerequisites

To complete this walkthrough, you should have:

1. An origin and destination network of choice
2. , between which you'd like to deploy the xERC20 warp route.
3. The address of an xERC20 or xERC20Lockbox
4. on the origin and destination chain.
5. Hyperlane CLI installed
6. and a wallet private key set as the HYP_KEY
7. environment variable
8. , funded on both origin and destination networks.

Step 1: Initialize a New Warp Route

Run the following command to generate a warp route configuration:

`hyperlane warp init` 1. Select Ethereum 2. (origin) and YourChain 3. (destination) using space, then hit enter. 4. For Ethereum (origin), choose xERC20 5. or xERC20Lockbox 6. , and enter the contract address. 7. For YourChain (destination), choose xERC20 8. or xERC20Lockbox 9. , and enter the contract address.

info Choosing Between xERC20 and xERC20Lockbox:

- xERC20Lockbox
 - : Use this if your token is already deployed on the origin chain and requires a wrapped version for cross-chain transfers.
- xERC20
 - : Use this if your token natively implements xERC20 and does not require a lockbox.

If you're unsure, review your token's contract structure or deployment history to determine the correct choice.

Step 2: Deploy the xERC20 Warp Route

Run the following command to deploy the warp route:

`hyperlane warp deploy` This deploys the Hyperlane adapter contracts *, either:

- HypXERC20Lockbox
- (if using xERC20Lockbox)
- HypXERC20
- (if using xERC20)

The CLI does not deploy non-Hyperlane contracts (e.g., existing xERC20 contracts). Those should already be deployed.

Step 3: Configure Minting Permissions

Once the xERC20 warp route contracts are deployed, you must configure minting permissions for the destination token.

Set the mint limit for the newly deployed warp route using the [setLimits function](#) in your xERC20 contract. By default, these limits are found in the Registry, located at `~/hyperlane` locally.

Step 4: Test Cross-Chain Transfers

Test transferring tokens across chains using:

`hyperlane warp send --symbol`

`< token_symbol`

`--amount`

`< amount`

- Replace

- with your token.
- Replace
- with the number of tokens to transfer.

Congrats! You have successfully created an xERC20 Warp Route

. Your xERC20 tokens can now move across chains!

warning Please note that cross-chain operations rely on Interchain Security Modules (ISMs) , and there are inherent risks, such as ISM trust assumptions.

Resources

For more resources on deploying xERC20 warp routes, check out:

- [xERC20 Token Standard](#)
- [xERC20 Token transfer limits](#) [Edit this page](#) [Previous](#) [How to extend an existing Warp Route](#) [Next](#) [Deploy an SVM Warp Route](#)