

# Register a Custom Logic Upkeep

Create powerful automation for your smart contract that leverages custom logic to trigger specified actions. This guide explains how to register a custom logic upkeep that uses [a compatible contract](#). You can register it using the Chainlink Automation App or from within a contract that you deploy.

Testing and best practices

Follow the [best practices](#) when creating a compatible contract and test your upkeep on a testnet before deploying it to a mainnet.

## Using the Chainlink Automation App

[Open the Chainlink Automation App](#) Click the Register New Upkeep button

### Connecting your wallet

If you do not already have a wallet connected with the Chainlink Automation network, the interface will prompt you to do so. Click the [Connect Wallet](#) button and follow the remaining prompts to connect your wallet to one of the [Automation supported blockchain networks](#).

### Trigger selection

Select Custom Logic trigger.

### Using custom logic triggers

Provide the address of your [compatible contract](#). You do not need to verify the contract onchain, but it must be [compatible](#) with the `AutomationCompatibleInterface` contract.

### Entering upkeep details

Provide the following information in the Automation app:

- Upkeep name: This will be publicly visible in the Chainlink Automation app.
- Gas limit: This is the maximum amount of gas that your transaction requires to execute on chain. This limit cannot exceed the `performGasLimit` value configured on the [registry](#). Before the network executes your transaction on chain, it simulates the transaction. If the gas required to execute your transaction exceeds the gas limit that you specified, your transaction will not be confirmed. Developers also have the ability to update `performGasLimit` for an upkeep. Consider running your function on a testnet to see how much gas it uses before you select a gas limit. This can be changed afterwards.
- Starting balance (LINK): Specify a LINK starting balance to fund your upkeep. See the [LINK Token Contracts](#) page to find the correct contract address and access faucets for testnet LINK. This field is required. You must have LINK before you can use the Chainlink Automation service.

Funding Upkeep

You should fund your contract with more LINK than you anticipate you will need. The network will not check or perform your Upkeep if your balance is too low based on current exchange rates. View the [Automation economics](#) page to learn more about the cost of using Chainlink Automation. [ERC-677 Link](#)

For funding on Mainnet, you need ERC-677 LINK. Many token bridges give you ERC-20 LINK tokens. Use PegSwap to [convert Chainlink tokens \(LINK\) to be ERC-677 compatible](#). To fund on a supported testnet, get [LINK](#) from [faucets.chain.link](#). \* Check data: This field is provided as an input for when your `checkUpkeep` function is simulated. Either leave this field blank or specify a hexadecimal value starting with 0x. To learn how to make flexible upkeeps using `checkData`, see the [Flexible Upkeeps](#) guide. \* Your email address (optional): This email address will be used to send you an email notification when your upkeep is underfunded.

### Complete upkeep registration

Click [Register](#) and confirm the transaction in MetaMask.

Your upkeeps will be displayed in your list of [Active Upkeeps](#). You must monitor the balance of your upkeep. If the balance drops below the minimum balance, the Chainlink Automation Network will not perform the Upkeep. See [Managing Upkeeps](#) to learn how to manage your upkeeps.