

Getting Started with Chainlink Automation

Chainlink Automation will reliably execute smart contract functions using a variety of triggers. Explore the examples below to see how Chainlink Automation works for each type of trigger. Before you begin, you will need an active cryptocurrency wallet such as Metamask.

- Time-based trigger: Use a time-based trigger to execute your function according to a time schedule.
- Custom logic trigger: Use a custom logic trigger to provide custom solidity logic that Automation Nodes evaluate (offchain) to determine when to execute your function onchain.
- Log trigger: Use log data as both trigger and input.

[Try out Chainlink Automation](#)

Click the tabs below to use Chainlink Automation with each type of trigger:

Time-basedCustom LogicLog TriggerIncrement a counter every 5 minutes using our example contract.

1. Navigate to the[Chainlink Automation app](#) and connect toArbitrum Sepoliain the top dropdown menu.
2. Connect your cryptocurrency wallet to the app if you haven't done so already. You may also need to fetch Arbitrum Sepolia testnet LINK[here](#).
3. ClickRegister new Upkeepand selectTime-basedtrigger.
4. UnderTarget contract address, enter0x083935210524c0A8922ec610d1063Aa0A54d9d70. This is a simple counter contract that increments with each call. View the source code[here](#).
5. In theContract callsection, enteraddIntegerunderTarget function. In theFunction inputssection, enter a number to increment by underintToAdd. Then clickNext.
6. Specify the time schedule, for example every 5 minutes. Paste the cron expression*/5 * * * *underCron expressionor select one of the example timers. Then clickNext.
7. To learn more about CRON expressions, click[here](#).
8. Enter anUpkeep name, your public key address underAdmin Address,500000underGas limit, and0.1underStarting balance (LINK).
9. ClickRegister Upkeep.
10. After the transaction has completed, you can view the performs for your upkeep in the upkeep details.

You have successfully automated your first time-based upkeep. To learn more about creating time-based upkeeps, read[here](#).Increment a counter using custom logic stored onchain.

1. Navigate to the[Chainlink Automation app](#) and connect toArbitrum Sepoliain the top dropdown menu.
2. Connect your cryptocurrency wallet to the app if you haven't done so already. You might also need to fetch LINK for the Arbitrum Sepolia testnet from[faucets.chain.link](#).
3. ClickRegister new Upkeepand selectCustom logictrigger.
4. UnderTarget contract address, enter0x6C0AAAEBCDb6F5D03759B8BF14b47BE491755530. This contract is an Automation-compatible contract that uses logic stored onchain and onchain state to determine when to increment a counter. View the source code[here](#). ClickNext.
5. Enter anUpkeep name, your public key address underAdmin Address,500000underGas limit, and0.1underStarting balance (LINK).
6. Finally, enter your public key address underCheck data (Hexadecimal).checkDatais optional static data that you can pass into your upkeep to ensure your counter increments.
7. ClickRegister Upkeep.
8. After the transaction is complete, you can view the performs for your upkeep in the upkeep details. Your upkeep should perform once every minute and stop after 4 performs.

You have successfully automated your first custom logic upkeep. To learn more about creating custom logic upkeeps, read[here](#).Increment an onchain counter using a log as trigger.

1. Navigate to the[Chainlink Automation app](#) and connect toArbitrum Sepoliain the top dropdown menu.
2. Connect your cryptocurrency wallet to the app if you haven't done so already. You might also need to fetch LINK for the Arbitrum Sepolia testnet from[faucets.chain.link](#).
3. ClickRegister new Upkeepand selectLog trigger.
4. UnderContract to automate, enter0xe817e4A71C69C72C01B31906F9F8591FbaB6b448. This is a simpleLogAutomation-compatible example contract that increments a counter when a log is detected. View the source code[here](#). ClickNext.
5. UnderContract emitting logs, enter0x1260206b960bB07F12d48C19fad505CeFc071bDd. This is the contract Automation will listen to for emitted logs. View the source code[here](#). ClickNext.
6. UnderEmitted logselectBumpfrom the dropdown menu. This is the log signature Automation will look for.
7. Log index topic filtersare optional filters to narrow the logs you want to trigger your upkeep. For this example, enter your public key address underaddrand leave thenumfield empty. Later when you call thebumpfunction to emit the log, yourmsg.Senderaddress will be emitted in the log, triggering your upkeep. ClickNext.
8. Enter anUpkeep name, your public key address underAdmin Address,500000underGas limit, and0.1underStarting

balance (LINK).

9. Click Register Upkeep and wait for the transaction to complete.
10. To trigger your upkeep call bump on the trigger contract by navigating to the Arbitrum Sepolia [scanner](#), connecting your wallet and executing the bump function. You can observe your upkeep's performance in the Automation dashboard.

You have successfully automated your first log trigger upkeep. To learn more about creating log trigger upkeeps, read [here](#).

Supported networks and costs

For a list of blockchains that are supported by Chainlink Automation, see the [Supported Networks](#) page. To learn more about the cost of using Chainlink Automation, see the [Automation Economics](#) page.

Contact us

For help with your specific use case [contact us](#) to connect with one of our Solutions Architects. You can also ask questions about Chainlink Automation on [Stack Overflow](#) or the [#automation channel](#) in our [Discord server](#). [Utility contracts](#) can also help you get started quickly.