

Automation Concepts

Before you explore how Chainlink Automation works on the [architecture](#) page, you should explore core concepts.

Prerequisites:

- [Smart contracts](#)
- [ERC-677 Token Standard](#)

Upkeeps and triggers

These are the jobs or tasks that you execute onchain. For example, you can call a smart contract function if a specific set of conditions are met. These specific conditions are called triggers. There are currently three types of triggers that the Chainlink Automation Network supports including:

- [Time-based trigger](#) : Use a [time-based trigger](#) to execute your function according to a time schedule. This feature is also called the Job Scheduler and it is a throwback to the Ethereum Alarm Clock. Time-based trigger contracts do not need to be [compatible](#) with the `AutomationCompatibleInterface` contract.
- [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. Your contract must meet the requirements to be [compatible](#) with the `AutomationCompatibleInterface` contract. Custom logic examples include checking the balance on a contract, only executing limit orders when their levels are met, any one of our [coded examples](#) , and many more.
- [Log trigger](#) : Use log data as both trigger and input. Your contract must meet the requirements to be [compatible](#) with the `AutomationCompatibleInterface` contract.

Automation nodes

Automation Nodes in the Chainlink Automation Network provide service to upkeeps that are funded and registered in the Automation registry. Automation Nodes use the same Node Operators as Chainlink Data Feeds.