

# Data Feeds on Solana

Chainlink provides data feeds on the [Solana](#) network. Chainlink data feeds on Solana employ [Offchain Reporting \(OCR\)](#) to aggregate data from data providers who pull from both centralized and decentralized exchanges. Chainlink's Solana deployment has no dependencies on external blockchain networks such as Ethereum. In Solana, storage and smart contract logic are separate. Programs store all the logic similar to an EVM (Ethereum) smart contract. The accounts store all the data. Compared to Solidity, the combination of an account and a program is equivalent to a smart contract on an EVM chain. State and logic are separate in Solana.

Solana programs are stateless, so you don't always need to deploy your program to the network to test it. You can deploy and test your programs on a [Solana Test Validator](#) . However, to use Chainlink products on Solana, you must deploy your contract onchain to one of the [supported Solana clusters](#) .

Note

Please note that Price Feeds performance relies on the chains they are deployed on. Periods of high network congestion may impact the frequency of Chainlink Price Feeds. Subscribe to [Solana status](#) notifications to stay updated on system performance.

To learn how to mitigate risk to your applications, read the [Selecting Quality Data Feeds](#) page.

## [Chainlink products and Solana clusters](#)

[Price Feeds](#) are available on the following Solana clusters:

- [Solana Mainnet](#)
- [Solana Devnet](#)

Solana provides a [Testnet cluster](#) that runs newer [Solana releases](#) , but Chainlink Data Feeds are not available on this cluster.

See the [Solana Data Feeds](#) page for a full list of Chainlink data feeds that are available on Solana.

To learn when more Chainlink services become available, follow us on [Twitter](#) or sign up for our [mailing list](#) .

## [Languages, tools, and frameworks](#)

The examples in the Chainlink documentation use the following languages, tools, and frameworks:

- [Node.js 14 or higher](#) : Used to run client code
- [Rust](#) : A general-purpose programming language designed for performance and memory safety
- [Anchor](#) : A framework for the [Solana Sealevel runtime](#) that provides several developer tools
- [Chainlink Solana Starter Kit](#) : An Anchor based program and client that shows developers how to use and interact with Chainlink Data Feeds on Solana
- [Solana CLI](#) : The Solana command line interface
- [Git](#) : Used to clone the example code repository

When developing applications to use Chainlink products on Solana, always use [Mainnet release](#) version of the Solana CLI that is equal to or greater than the version currently running on your target cluster. Use `solana --version` and `solana cluster-version` to check CLI and cluster versions:

```
solana--version
solana-cli1.9.28(src:b576e9cc;feat:320703611)
solana cluster-version--urldevnet1.9.25
```

```
solana cluster-version--urlmainnet-beta1.9.28
```

The examples in this documentation use Solana programs in [Rust](#) , but you can also write Solana programs in [C](#) . To learn more about the Solana programming model, see the [Solana Documentation](#) .

## [Solana wallets](#)

When you use Chainlink on Solana, you need a [Solana wallet](#) . The Chainlink documentation uses [file system wallets](#) and free Devnet SOL tokens to demonstrate examples. When you deploy your programs to the Solana Mainnet, you must use wallets with mainnet lamports.

If you have existing wallets that you want to use for the guides in the Chainlink documentation, find your wallet keypair and make it available in your development environment as a file. You can point [Anchor](#) and the [Solana CLI](#) to a specific keypair when you deploy or manage your Solana programs.

`anchor build` :

`anchor deploy--provider.wallet~/.config/solana/id.json--provider.clusterdevnet` :

`solana program show--programs--keypair~/.config/solana/id.json--urldevnet`

Program Id|Slot|Authority|Balance

6U4suTp55kiJRKqV7HGAQvFgcLaStLnUA4myg5DRqsKw|109609728|E6gKKToCJPgf4zEL1GRLL6T99g2WcfAzJAMvtma1KijT|2.57751768SOL

When you build your production applications and deploy Solana programs to the Mainnet cluster, always follow the security best practices in the [Solana Wallet Guide](#) for managing your wallets and keypairs.