

Setup a Solana RPC Node

Since a Solana RPC server runs the same process as a consensus validator, first follow the instructions on [how to setup a Solana validator](#) to get started. Note, that you do not need to create a vote account if you are operating an RPC node. An RPC node typically does not vote.

After your validator is running, you can refer to this section for the RPC node specific setup instructions.

Sample RPC Node

Below is an example `validator.sh` file for atestnet RPC server.

You will want to be aware of the following flags:

- `--full-rpc-api`
 - : enables all RPC operations on this validator.
- `--no-voting`
 - : runs the validator without participating in consensus. Typically, you do not want to run a validator as both a consensus node and a full RPC node due to resource constraints.
- `--private-rpc`
 - : does not publish the validator's open RPC port in the solana gossip
- `command`

For more explanation on the flags used in the command, refer to `thesolana-validator --help` command

#!/bin/bash

```
exec solana-validator \ --identity /home/sol/validator-keypair.json \ --known-validator
5D1fNXzvv5NjV1ysLjirC4WY92RNsVH18vjmcSzZd8on \ --known-validator
dDzy5SR3AXdYWWqbDEkVFdvSPCtS9ihF5kJKHctXoFs \ --known-validator
eoKpUABi59aT4rR9HGS3LcMecfut9x7zJyodWWP43YQ \ --known-validator
7XSY3MrYnK8vq693Rju17bbPkCN3Z7KvfvJx4kdrsSY \ --known-validator
Ft5fbkqNa76vnsjYNwjDZUXoTWpP7VYm3mtsaQckQADN \ --known-validator
9QxCLckBiJc783jnMvXZubK4wH86Eqqvashtwvcsgkv \ --only-known-rpc \ --full-rpc-api \ --no-voting \ --ledger /mnt/ledger \ -
-accounts /mnt/accounts \ --log /home/sol/solana-rpc.log \ --rpc-port 8899 \ --rpc-bind-address 0.0.0.0 \ --private-rpc \ --
dynamic-port-range 8000-8020 \ --entrypoint entrypoint.testnet.solana.com:8001 \ --entrypoint
entrypoint2.testnet.solana.com:8001 \ --entrypoint entrypoint3.testnet.solana.com:8001 \ --expected-genesis-hash
4uhcVJyU9pJkvQyS88uRDiswHXSckY3zQawwpjk2NsNY \ --wal-recovery-mode skip_any_corrupted_record \ --limit-ledger-
size
```

Solana Bigtable

The Solana blockchain is able to create many transactions per second. Because of the volume of transactions on the chain, it is not practical for an RPC node to store the entire blockchain on the machine. Instead, RPC operators use the `--limit-ledger-size` flag to specify how many blocks to store on the RPC node. If the user of the RPC node needs historical blockchain data then the RPC server will have to access older blocks through a Solana bigtable instance.

If you are interested in setting up your own bigtable instance, see these docs in the Solana GitHub repository [solana-labs/solana-bigtable](#)

Example Known Validators

The identities of the [known validators](#) supplied in these example snippets (via the `--known-validator` flag) are:

- 5D1fNXzvv5NjV1ysLjirC4WY92RNsVH18vjmcSzZd8on
 - Solana Labs
- dDzy5SR3AXdYWWqbDEkVFdvSPCtS9ihF5kJKHctXoFs
 - MonkeDAO
- Ft5fbkqNa76vnsjYNwjDZUXoTWpP7VYm3mtsaQckQADN
 - Certus One
- eoKpUABi59aT4rR9HGS3LcMecfut9x7zJyodWWP43YQ
 - SerGo

- 9QxCLckBiJc783jnMvXZubK4wH86Eqqvashtwvcsgkv
- - Algo|Stake

Examples for other clusters

Additional examples of other Solana cluster specific validator commands can be found on the [Clusters](#) page.

Keep in mind, you will still need to customize these commands to operate as an RPC node, as well other operator specific configuration settings.

Account indexing

As the number of populated accounts on the cluster grows, account-data RPC requests that scan the entire account set -- like [getProgramAccounts](#) and [SPL-token-specific requests](#) -- may perform poorly. If your validator needs to support any of these requests, you can use the `--account-index` parameter to activate one or more in-memory account indexes that significantly improve RPC performance by indexing accounts by the key field. Currently supports the following parameter values:

- `program-id`
- `:` each account indexed by its owning program; used by [getProgramAccounts](#)
- `spl-token-mint`
- `:` each SPL token account indexed by its token Mint; used by [getTokenAccountsByDelegate](#)
- `,` and [getTokenLargestAccounts](#)
- `spl-token-owner`
- `:` each SPL token account indexed by the token-owner address; used by [getTokenAccountsByOwner](#)
- `,` and [getProgramAccounts](#)
- requests that include an `spl-token-owner` filter. [Previous Setup a Validator Next Best Practices: Validator Operations](#)