Validator Guide: Vote Account Management

This page describes how to set up an on-chainvote account . Creating a vote account is needed if you plan to run a validator node on Solana.

Create a Vote Account

A vote account can be created with thecreate-vote-account command. The vote account can be configured when first created or after the validator is running. All aspects of the vote account can be changed except for thevote account address , which is fixed for the lifetime of the account.

Configure an Existing Vote Account

- To change thevalidator identity
- , usevote-update-validator
- .
- To change thevote authority
- , usevote-authorize-voter-checked
- .
- To change theauthorized withdrawer
- , usevote-authorize-withdrawer-checked
- .
- To change thecommission
- , usevote-update-commission
- .

Vote Account Structure

Vote Account Address

A vote account is created at an address that is either the public key of a keypair file, or at a derived address based on a keypair file's public key and a seed string.

The address of a vote account is never needed to sign any transactions, but is just used to look up the account information.

When someone wants todelegate tokens in a stake account, the delegation command is pointed at the vote account address of the validator to whom the token-holder wants to delegate.

Validator Identity

Thevalidator identity is a system account that is used to pay for all the vote transaction fees submitted to the vote account. Because the validator is expected to vote on most valid blocks it receives, the validator identity account is frequently (potentially multiple times per second) signing transactions and paying fees. For this reason the validator identity keypair must be stored as a "hot wallet" in a keypair file on the same system the validator process is running.

Because a hot wallet is generally less secure than an offline or "cold" wallet, the validator operator may choose to store only enough SOL on the identity account to cover voting fees for a limited amount of time, such as a few weeks or months. The validator identity account could be periodically topped off from a more secure wallet.

This practice can reduce the risk of loss of funds if the validator node's disk or file system becomes compromised or corrupted.

The validator identity is required to be provided when a vote account is created. The validator identity can also be changed after an account is created by using thevote-update-validator command.

Vote Authority

Thevote authority keypair is used to sign each vote transaction the validator node wants to submit to the cluster. This doesn't necessarily have to be unique from the validator identity, as you will see later in this document. Because the vote authority, like the validator identity, is signing transactions frequently, this also must be a hot keypair on the same file system as the validator process.

The vote authority can be set to the same address as the validator identity. If the validator identity is also the vote authority, only one signature per vote transaction is needed in order to both sign the vote and pay the transaction fee. Because transaction fees on Solana are assessed per-signature, having one signer instead of two will result in half the transaction fee paid compared to setting the vote authority and validator identity to two different accounts.

The vote authority can be set when the vote account is created. If it is not provided, the default behavior is to assign it the same as the validator identity. The vote authority can be changed later with thevote-authorize-voter-checked command.

The vote authority can be changed at most once per epoch. If the authority is changed withvote-authorize-voter-checked , this will not take effect until the beginning of the next epoch. To support a smooth transition of the vote signing,solana-validator allows the--authorized-voter argument to be specified multiple times. This allows the validator process to keep voting successfully when the network reaches an epoch boundary at which the validator's vote authority account changes.

Authorized Withdrawer

Theauthorized withdrawer keypair is used to withdraw funds from a vote account using thewithdraw-from-vote-account command. Any network rewards a validator earns are deposited into the vote account and are only retrievable by signing with the authorized withdrawer keypair.

The authorized withdrawer is also required to sign any transaction to change a vote account'scommission , and to change the validator identity on a vote account.

Because theft of an authorized withdrawer keypair can give complete control over the operation of a validator to an attacker, it is advised to keep the withdraw authority keypair in an offline/cold wallet in a secure location. The withdraw authority keypair is not needed during operation of a validator and should not stored on the validator itself.

The authorized withdrawer must be set when the vote account is created. It must not be set to a keypair that is the same as either the validator identity keypair or the vote authority keypair.

The authorized withdrawer can be changed later with thevote-authorize-withdrawer-checked command.

Commission

Commission is the percent of network rewards earned by a validator that are deposited into the validator's vote account. The remainder of the rewards are distributed to all of the stake accounts delegated to that vote account, proportional to the active stake weight of each stake account.

For example, if a vote account has a commission of 10%, for all rewards earned by that validator in a given epoch, 10% of these rewards will be deposited into the vote account in the first block of the following epoch. The remaining 90% will be deposited into delegated stake accounts as immediately active stake.

A validator may choose to set a low commission to try to attract more stake delegations as a lower commission results in a larger percentage of rewards passed along to the delegator. As there are costs associated with setting up and operating a validator node, a validator would ideally set a high enough commission to at least cover their expenses.

Commission can be set upon vote account creation with the--commission option. If it is not provided, it will default to 100%, which will result in all rewards deposited in the vote account, and none passed on to any delegated stake accounts.

Commission can also be changed later with thevote-update-commission command.

When setting the commission, only integer values in the set[0-100]are accepted. The integer represents the number of percentage points for the commission, so creating an account with--commission 10 will set a 10% commission.

Note that validators can only update their commission during the first half of any epoch. This prevents validators from stealing delegator rewards by setting a low commission, increasing it right before the end of the epoch, and then changing it back after reward distribution.

Key Rotation

Rotating the vote account authority keys requires special handling when dealing with a live validator.

Note that vote account key rotation has no effect on the stake accounts that have been delegated to the vote account. For example it is possible to use key rotation to transfer all authority of a vote account from one entity to another without any impact to staking rewards.

Vote Account Validator Identity

You will need access to theauthorized withdrawer keypair for the vote account to change the validator identity. The following steps assume that~/authorized_withdrawer.json is that keypair.

1. Create the new validator identity keypair,solana-keygen new -o ~/new-validator-keypair.json
2. .
3. Ensure that the new identity account has been funded,solana transfer ~/new-validator-keypair.json 500
4. .
5. Runsolana vote-update-validator ~/vote-account-keypair.json ~/new-validator-keypair.json

~/authorized_withdrawer.json
6. to modify the validator identity in your vote account
7. Restart your validator with the new identity keypair for the--identity
8. argument

Additional steps are required if your validator has stake. The leader schedule is computed two epochs in advance. Therefore if your old validator identity was in the leader schedule, it will remain in the leader schedule for up to two epochs after the validator identity change. If extra steps are not taken your validator will produce no blocks until your new validator identity is added to the leader schedule.

After your validator is restarted with the new identity keypair, per step 4, start a second non-voting validator on a different machine with the old identity keypair without providing the--vote-account argument, as well as with the--no-wait-for-vote-to-start-leader argument.

This temporary validator should be run for two full epochs. During this time it will:

- Produce blocks for the remaining slots that are assigned to your old validator
- identity
- Receive the transaction fees and rent rewards for your old validator identity

It is safe to stop this temporary validator when your old validator identity is no longer listed in thesolana leader-schedule output.

Vote Account Authorized Voter

Thevote authority keypair may only be changed at epoch boundaries and requires some additional arguments tosolana-validator for a seamless migration.

1. Runsolana epoch-info
2. . If there is not much time remaining time in the
3. current epoch, consider waiting for the next epoch to allow your validator
4. plenty of time to restart and catch up.
5. Create the new vote authority keypair,solana-keygen new -o ~/new-vote-authority.json
6. .
7. Determine the currentvote authority
8. keypair by runningsolana vote-account ~/vote-account-keypair.json
9. . It may be validator's
10. identity account (the default) or some other keypair. The following steps
11. assume that~/validator-keypair.json
12. is that keypair.
13. Runsolana vote-authorize-voter-checked ~/vote-account-keypair.json ~/validator-keypair.json ~/new-vote-authority.json
14. .
15. The new vote authority is scheduled to become active starting at the next
16. epoch.
17. solana-validator
18. now needs to be restarted with the old and new vote
19. authority keypairs, so that it can smoothly transition at the next epoch. Add
20. the two arguments on restart:--authorized-voter ~/validator-keypair.json --authorized-voter ~/new-vote-authority.json
21. After the cluster reaches the next epoch, remove the--authorized-voter ~/validator-keypair.json
22. argument and restartsolana-validator
23. , as the old vote authority keypair is no longer required.

Vote Account Authorized Withdrawer

No special handling or timing considerations are required. Use thesolana vote-authorize-withdrawer-checked command as needed.

Consider Durable Nonces for a Trustless Transfer of the Authorized Voter or Withdrawer

If the Authorized Voter or Withdrawer is to be transferred to another entity then a two-stage signing process using a Durable Nonce is recommended.

1. Entity B creates a durable nonce usingsolana create-nonce-account
2. Entity B then runs asolana vote-authorize-voter-checked
3. orsolana vote-authorize-withdrawer-checked

4. command, including:

5. the--sign-only

6. argument
7. the--nonce
8. ,--nonce-authority
9. , and--blockhash
10. arguments to specify the
11. nonce particulars
12. the address of the Entity A's existing authority, and the keypair for Entity
13. B's new authority
14. When thesolana vote-authorize-...-checked
15. command successfully executes,
16. it will output transaction signatures that Entity B must share with Entity A
17. Entity A then runs a similarsolana vote-authorize-voter-checked
18. orsolana vote-authorize-withdrawer-checked
19. command with the following
20. changes:
21. the--sign-only
22. argument is removed, and replaced with a--signer
23. argument
24. for each of the signatures provided by Entity B
25. the address of Entity A's existing authority is replaced with the
26. corresponding keypair, and the keypair for Entity B's new authority is
27. replaced with the corresponding address

On success the authority is now changed without Entity A or B having to reveal keypairs to the other even though both entities signed the transaction.

Close a Vote Account

A vote account can be closed with theclose-vote-account command. Closing a vote account withdraws all remaining SOL funds to a supplied recipient address and renders it invalid as a vote account. It is not possible to close a vote account with active stake. Previous Validator Guides: Troubleshooting Next Validator Guides: Node Failover