Running a relayer

The relayer can be run on a separate (smaller) machine than the validator, or co-hosted on the same machine. A few settings below are different for co-hosted vs. separate host setups.

Note:There is no reason to run a relayer unless you are connecting to a block-producing validator (on the leader schedule).

Preparation

The relayer needs to authenticate connections from validators. It accomplishes this through challenge/reponse system that creates authentication tokens signed by both the relayer and validator. The relayer uses a pair of JWT tokens that can be generated with the following commands:

```
Copy OUTPUT_DIR= openssl genrsa --out OUTPUT_DIR/private.pem openssl rsa --in OUTPUT_DIR/private.pem --pubout --out OUTPUT_DIR/public.pem
```

Fill in OUTPUT_DIR with the path where you would like to store the tokens. You will need to supply this path in your relayer startup below.

The relayer also needs to be authenticated by the block engine. For this, you need to generate a solana keypair and open a ticket in the Jito Discord to get whitelisted to block engines:

```
Copy solana-keygen new --no-bip39-passphrase --outfile relayer-keypair.json
```

RPC Server

The relayer needs access to an rpc to for two purposes

- determine when to forward packets (leader schedule and slot)
- keep track of account lookup tables for transaction forwarding
-

If you're running a co-hosted setup (same machine as validator), then you can use your validator as the rpc (add--private-rpc ,--full-rpc-api , and--rpc-port 8899 to your validator startup). If you're running on a separate machine, then you will need to supply a list (see relayer arguments below) of URLs for rpcs. We recommend more than 1 rpc for redundancy. The relayer will automatically load balance between rpcs by choosing the one that currently returns the highest slot.

Note:

For the best performance, you must have an index enabled on the rpc for the Address Lookup Table Program. If you're not indexing on program-id, or want to have the minimal set of indices, you can add the arguments to your RPC file:

```
Copy --account-index program-id \ --account-index-include-key AddressLookupTab1e1111111111111111111111111 \
```

Time Synchronization

It is important that the clock on your relayer machine is synchronized with that of our block engine servers. We run several ntp servers. Please find the one for your regionhere , and connect using an ntp client (we recommend chrony). An example chrony systemd service file and config are given below. There are numerous guides for running ntp clients (e.g.https://www.redhat.com/sysadmin/chrony-time-services-linux).

```
Copy
```

Example chrony.conf file for connectiong to jito ntp server

server ntp.tokyo.jito.wtf iburst logdir /var/log/chrony log statistics measurements tracking driftfile /var/lib/chrony/drift makestep 1.0 3 maxupdateskew 100.0 dumpdir /var/lib/chrony rtcsync// Some code

```
```

Running the Relayer

You need to either build or download the binary executable for the relayer. Instructions can be found in the relayer repository -https://github.com/jito-foundation/jito-relayer/tree/master .

We recommend running the relayer as aSystemd service. Two example systemd configuration files are shown below. Make sure to:

- replacePATH_TO_KEYS
- with the ones you generated above
- replace theUSER
- with your system user name
- supply theBLOCK_ENGINE_URL
- of your choice (SeeThird Party Block Engines
-)
- changeRELAYER_PATH
- to the path for your relayer executable and make sure that it has executable permissions
-

If you're running the separate host setup, you also need to supply rpc urls.

Note: You must setRUST_LOG=info in order to emit info level datapoints to the metrics server.

Co-Hosted Setup

```
```

Copy

Example Systemd File for Co-Hosted Relayer

[Unit] Description=Solana transaction relayer Requires=network-online.target #NTP Service connected to Jito NTP server (e.g. chrony) After=network-online.target #NTP Service connected to Jito NTP server (e.g. chrony)

User is required to install a keypair here that's used to auth against the block engine

ConditionPathExists={PATH_TO_KEYS}/id.json ConditionPathExists={PATH_TO_KEYS}/private.pem ConditionPathExists={PATH_TO_KEYS}/public.pem

[Service] Type=exec User={USER} Restart=on-failure Environment=RUST_LOG=info Environment=SOLANA_METRICS_CONFIG="host=http://metrics.jito.wtf:8086,db=relayer,u=relayer-operators,p=jito-relayer-write" Environment=BLOCK_ENGINE_URL={BLOCK_ENGINE_URL} Environment=GRPC_BIND_IP=127.0.0.1

ExecStart={RELAYER_PATH}/jito-transaction-relayer \ --keypair-path=/etc/relayer/keys/id.json \ --signing-key-pem-path=/etc/relayer/keys/private.pem \ --verifying-key-pem-path=/etc/relayer/keys/public.pem

[Install] WantedBy=multi-user.target

```
```

```
```

Copy

Extra Solana Validator CLI Arguments for Co-Hosted Relayer

--relayer-url http://127.0.0.1:11226 --private-rpc --full-rpc-api --rpc-port 8899 --account-index program-id --account-index-include-key AddressLookupTab1e1111111111111111111111111

```

Separate Host Setup

```

Copy [Unit] Description=Solana transaction relayer Requires=network-online.target #NTP Service connected to Jito NTP server (e.g. chrony) After=network-online.target #NTP Service connected to Jito NTP server (e.g. chrony)

User is required to install a keypair here that's used to auth against the block engine

ConditionPathExists={PATH_TO_KEYS}/id.json ConditionPathExists={PATH_TO_KEYS}/private.pem ConditionPathExists={PATH_TO_KEYS}/public.pem

[Service] Type=exec User={USER} Restart=on-failure Environment=RUST_LOG=info Environment=SOLANA_METRICS_CONFIG="host=http://metrics.jito.wtf:8086,db=relayer,u=relayer-operators,p=jito-relayer-write" Environment=BLOCK_ENGINE_URL={BLOCK_ENGINE_URL} Environment=RPC_SERVERS=https://rpc1.your.domain/your_api_token https://rpc2.your.domain/your_api_token Environment=WEBSOCKET_SERVERS=wss://rpc1.your.domain/your_api_token wss://rpc2.your.domain/your_api_token

ExecStart={RELAYER_PATH}/jito-transaction-relayer \ --keypair-path=/etc/relayer/keys/id.json \ --signing-key-pem-path=/etc/relayer/keys/private.pem \ --verifying-key-pem-path=/etc/relayer/keys/public.pem

[Install] WantedBy=multi-user.target

```

Firewall Settings

You will need to open udp ports fortpu_quic_port andtpu_quic_forward_port on the relayer machine (default11228 and11229 ).

If you are running the relayer on a separate host, you will also need to open a tcp port forgrpc_bind_port default (11226 ). This is not needed for the co-hosted setup.

Monitoring

A public dashboard is available for viewing basic operational metrics for the relayer[https://grafana.metrics.jito.wtf:3000/](https://grafana.metrics.jito.wtf:3000/)

?

The relayer host name can be selected from the dropdown. The default host is Jito's co-hosted test setup and can be used as an example of what to expect for this configuration with a low staked validator.

- Relayer health is used to determine if the relayer's view of the current slot is up to date
- (1 = Good, 0 = Bad).
- The number of validators connected should be 1 for a co-hosted setup.
- The slot count shows the most up to date slot the relayer has received.
- TPU Packet Rate shows the rate (packets per second) of packets received and passing sigverify/
- Validator Packet Rate shows the rate (packets per second) of packets being sent to each validator.
- Block Engine Packet Rate shows the rate (packets per second) of packets being sent to the Block Engine.
-

Packet rates vary with time on the network, but they should be similar for validators with leader slots close in time.

Relayer Command Line Arguments

As seen in the systemd files above, the default values for most arguments should work for the majority of cases. However, the relayer is highly configurable based on individual nees. We have included descriptions of the most commonly configured arguments below. A full list of command line arguments can be found in the relayer source code ([https://github.com/jito-foundation/jito-relayer/blob/master/transaction-relayer/src/main.rs](https://github.com/jito-foundation/jito-relayer/blob/master/transaction-relayer/src/main.rs) )

Required arguments

block-engine-url : SeeThird Party Block Engines

keypair-path : The path to the keypair file used to authenticate with a block engine.
```

signing-key-pem-path : The private key used to sign tokens by this server.

verifying_key_pem_path : The public key used to verify tokens by this and other services.

Optional

rpc-servers : A space-separated list of RPC servers to connect to for network information. It's also reccomended that you run multiple RPC servers for redundancy purposes.

websocket-servers : A space-separted list of RPC servers to listen to for slot updates. Must be a 1:1 mapping in ordering as the rpc-servers command line argument.

grpc-bind-ip : Bind IP address for GRPC server that listens for validator connections.

public-ip : The public-facing IP of the relayer server. This is advertised on gossip as the place to send transactions to, and needs to be an ip that is routed the machine where the relayer is running. This is provided for situations where looking up the public ip through (ifconfig.me) is not possible.

allowed-validators : An allow-list for validators to connect to the relayer. If not specified, then any relayer on the leader schedule is allowed to connect.

Previous Features Next Reliability Last updated8 days ago On this page *Note: There is no reason to run a relayer unless you are connecting to a block-producing validator (on the leader schedule). * Preparation * RPC Server * Time Synchronization * Running the Relayer * Firewall Settings * Monitoring * Relayer Command Line Arguments *