

Transaction Family Lab Manual

Pre-requisites

- Install Docker Engine and Docker Compose.
- Download the Sawtooth Docker Compose file from Moodle ([sawtooth-default-blockinfo.yaml](#)).

Step 1: Start the Sawtooth Docker Environment

To start up the environment, perform the following tasks:

- Open a terminal window.
- Change your working directory to the same directory where you saved the Docker Compose file ([sawtooth-default-blockinfo.yaml](#)).
- Run the following command:

```
$ sudo docker-compose -f sawtooth-default-blockinfo.yaml up
```

Downloading the Docker images for the Sawtooth environment can take several minutes.

Step 2: Log Into The Client Container

- Log into the client container by opening a new terminal window (**ctrl+shift+t**) and running the following command. Note that [sawtooth-shell-default](#) specifies the client container name.

```
$ sudo docker exec -it sawtooth-shell-default bash
```

Step 3: Log Into The Validator Container

- Log into the validator container by opening a new terminal window (**ctrl+shift+t**) and running the following command. Note that `sawtooth-validator-default` specifies the validator container name.

```
$ sudo docker exec -it sawtooth-validator-default bash
```

1. Settings Transaction Family

The Settings Transaction Family provides a methodology for storing on-chain configuration settings.

Note: To run the commands in this section, use the terminal window for **validator container**.

- 1.1. You can set a setting with the `sawset proposal create` command. For example, the following command will set the maximum transactions allowed per block as 100.

```
# sawset proposal create --url http://rest-api:8008  
--key /root/.sawtooth/keys/my_key.priv  
sawtooth.validator.max_transactions_per_block='100'
```

- 1.2. You can list existing settings with the following command

```
# sawtooth settings list --url http://rest-api:8008
```

- 1.3. Now generate some keys.

```
# sawtooth keygen keyone
```

```
# sawtooth keygen keytwo
```

```
# sawtooth keygen keythree
```

- 1.4. Add some of these generated keys to the `authorized_keys` and make the `approval_threshold` as 2. By default, `authorized_keys` contains the key which created the genesis block and `approval_threshold` is 1.

```
# sawset proposal create --url http://rest-api:8008
--key root/.sawtooth/keys/my_key.priv
sawtooth.settings.vote.authorized_keys=$(cat
/root/.sawtooth/keys/my_key.pub),$(cat
/root/.sawtooth/keys/keyone.pub),$(cat
/root/.sawtooth/keys/keytwo.pub)
sawtooth.settings.vote.approval_threshold=2
```

- 1.5. Add a new setting by giving the following command

```
# sawset proposal create --url http://rest-api:8008
--key /root/.sawtooth/keys/my_key.priv
sawtooth.validator.max_transactions_per_block='300'
```

You cannot see the newly added setting in the settings list (`# sawtooth settings list --url http://rest-api:8008`) because this setting needs at least two votes.

You can view currently proposed settings that are not yet active by giving the following command :

```
# sawset proposal list --url http://rest-api:8008
```

This command will list the proposal id and the newly added setting.

- 1.6. To vote for a specific settings-change proposal, run the following command. (Obtain `<proposal_Id>` by giving the following command :

```
# sawset proposal list --url http://rest-api:8008 )
```

```
# sawset proposal vote --url http://rest-api:8008
--key /root/.sawtooth/keys/keyone.priv
<proposal_Id> accept
```

Run the following command to check the setting change.

```
# sawtooth settings list --url http://rest-api:8008
```

- 1.7. Now change the settings to the previous default values

```
# sawset proposal create --url http://rest-api:8008
```

```
--key root/.sawtooth/keys/my_key.priv
```

```
sawtooth.settings.vote.approval_threshold=1
```

```
# sawset proposal vote --url http://rest-api:8008
```

```
-k /root/.sawtooth/keys/keyone.priv <proposalId> accept
```

(Obtain <proposalId> by giving the following command :

```
# sawset proposal list --url http://rest-api:8008 )
```

2. IntegerKey Transaction Family

The IntegerKey Transaction Family simply sets, increments, and decrements the value of entries stored in a state dictionary. The `intkey` command starts the IntegerKey transaction processor, which provides functions that can be used to test deployed ledgers.

Note: To run the commands in this section, use the terminal window for **client container** .

- 2.1. The `intkey set` subcommand sets a key (name) to the specified value.

```
# intkey set MyKey 65 --url http://rest-api:8008
```

- 2.2. The `intkey show` subcommand displays the value of the specified key.

```
# intkey show MyKey --url http://rest-api:8008
```

The output looks like:

```
MyKey: 65
```

- 2.3. The `intkey inc` subcommand increments a key by the specified value.
`# intkey inc MyKey 5 --url http://rest-api:8008`
- 2.4. The `intkey dec` subcommand decrements a key by the specified value.
`# intkey dec MyKey 10 --url http://rest-api:8008`
- 2.5. The `intkey list` subcommand displays the value of all keys.
`# intkey list --url http://rest-api:8008`

3. BlockInfo Transaction Family

The Blockinfo transaction family provides a way for storing information about a configurable number of historic blocks.

Note: Run the following commands from the **validator container**:

- 3.1. BlockInfo transactions should be added to a block by a BlockInfo Injector.
`# sawset proposal create`
`--key /root/.sawtooth/keys/my_key.priv`
`--url http://rest-api:8008`
`sawtooth.validator.batch_injectors=block_info`
- 3.2. Add the validation rules to ensure that only one transaction of this type is included at the beginning of the block. On-chain validation rules are stored as a string in the setting key `sawtooth.validator.block_validation_rules`.

`# sawset proposal create`
`--key /root/.sawtooth/keys/my_key.priv`
`--url http://rest-api:8008`
`sawtooth.validator.block_validation_rules='NofX:1,`
`block_info;XatY:block_info,0;local:0'`

Here the rules specifies that only one BlockInfo transaction is included per block, that the transaction is at the beginning of the block, and that the transaction is signed by the same key that signed the block. (**NofX** : Only N transaction of transaction type X may be included in a block, **XatY** : A transaction of type X must be in the block at position Y, **local** : A transaction must be signed by the same key as the block.)

- 3.3. Run the following command in the **client container**

```
# intkey inc MyKey 5 --url http://rest-api:8008
```

Now check the block details at <http://localhost:8008/blocks> where you can see that last created block will contain two transactions, in which first one will be BlockInfo transaction.

Run the following command in the client container

```
# sawtooth block list --url http://rest-api:8008
```

4. Additional Exercise

- 4.1. The following command will set the permitted transaction families

```
# sawset proposal create --url http://rest-api:8008 --key  
/root/.sawtooth/keys/my_key.priv  
sawtooth.validator.transaction_families='[{"family":  
"sawtooth_settings", "version": "1.0"}, {"family": "block_info",  
"version": "1.0"} ]'
```

This sawset command won't allow the validator to accept transactions of the type intkey (IntegerKey transaction family). So the following intkey transaction will fail.

Run the following command in the client container

```
# intkey inc MyKey 5 --url http://rest-api:8008
```

The terminal window in which you ran the docker-compose command displays the following log messages .

```
sawtooth-validator-default | [2018-11-21 04:15:36.714 DEBUG   executor] failing transaction 7c917069ca278a8aa1ac2fa5e9fd5fa5ed23e167d87a2bde6fff6e52135ddd462ed0c5dcfe6facf2983a7e5c5a8f7b897aa860def70ec9f6471ee8e3e918ae61 of type (name=intkey,version=1.0) since it isn't required in the configuration
sawtooth-validator-default | [2018-11-21 04:15:36.716 DEBUG   publisher] Batch dc6802d303cca7eae550a270a8e2ecb56be6075dbb08a6ec1b9272fadcd16f576bc669126a739c717f980b2123ace0e3332bcf02c9785a3ab84e6cc0ffcb36ed invalid, not added to block.
sawtooth-validator-default | [2018-11-21 04:15:36.717 DEBUG   publisher] Abandoning block (4, S:, P:fcf2193e): no batches added
```

- 4.2. Now run the following command to set back the permitted transaction families. Run the following command in the validator container

```
# sawset proposal create --url http://rest-api:8008 --key
/root/.sawtooth/keys/my_key.priv
sawtooth.validator.transaction_families='[{"family":
"sawtooth_settings", "version": "1.0"}, {"family": "block_info",
"version": "1.0"}, {"family": "intkey", "version": "1.0"} ]'
```

Now you will be able to perform intkey transactions.

5. Stop the Sawtooth Environment

Use this procedure if you need to stop or reset the Sawtooth environment for any reason.

- 5.1. Enter **CTRL-c** from the window where you originally ran docker-compose
- 5.2. After all containers have shut down, run this docker-compose command:

```
$ docker-compose -f sawtooth-default-blockinfo.yaml down
```

Sawtooth Settings List

The following are some Sawtooth settings:

- `sawtooth.config.authorization_type`
 - To set authorization type, use command line option `sawtooth-validator --network-auth {trust|challenge}`
- `sawtooth.consensus.algorithm.name`
 - Pluggable consensus algorithm name. These include PoET, Devmode, `sawtooth-raft-engine`, and `sawtooth-pbft`. The default is `devmode` for Sawtooth 1.1 or earlier.
- `sawtooth.consensus.algorithm.version`
 - Consensus algorithm version. Currently 0.1 for PoET, Devmode, and `sawtooth-pbft`, and 0.1.0 for `sawtooth-raft-engine`.
- `sawtooth.consensus.block_validation_rules`
 - Lists validation rules to use in deciding what blocks to add to the blockchain.
- `sawtooth.consensus.max_wait_time`
 - Maximum devmode consensus wait time, in seconds
- `sawtooth.consensus.min_wait_time`
 - Minimum devmode consensus wait time, in seconds
- `sawtooth.consensus.pbft.peers`
 - JSON list of each peer node's public key. Only required PBFT setting.
- `sawtooth.consensus.pbft.block_duration`
 - How often to try to publish a block. Optional, default 200 ms.
- `sawtooth.consensus.pbft.exponential_retry_base`
 - Base time to use for retrying with exponential backoff
Optional, default 100 ms
- `sawtooth.consensus.pbft.exponential_retry_max`
 - Maximum time for retrying with exponential backoff.
Optional, default 60s
- `sawtooth.consensus.pbft.faulty_primary_timeout`

- How long to wait for the next (BlockNew + PrePrepare) before determining primary is faulty. Should be $>$ block_duration. Optional, default 30s
- sawtooth.consensus.pbft.view_change_duration
 - If the node starts a change to view ($v + 2$), the timeout will be $(2 * \text{view_change_duration})$ Optional, default 5s
- sawtooth.consensus.pbft.forced_view_change_period
 - How many blocks to commit before forcing a view change for fairness Optional, default 30 blocks
- sawtooth.consensus.pbft.message_timeout
 - How long to wait for updates from the Consensus API. Optional, default 10 ms.
- sawtooth.consensus.pbft.max_log_size
 - Log size. Optional, default 1000 messages
- sawtooth.consensus.raft.election_tick
 - RAFT consensus election tick, in seconds. E.g., 1500
- sawtooth.consensus.raft.heartbeat_tick
 - RAFT consensus heartbeat tick, in seconds. E.g., 150
- sawtooth.consensus.raft.peers
 - JSON list of each peer node's public key. Only required RAFT setting.
- sawtooth.consensus.raft.period
 - RAFT consensus period, in seconds. E.g., 3. Higher settings cause larger blocks, small settings have faster performance with smaller, quicker block publication, but causes more network traffic.
- sawtooth.consensus.valid_block_publishers
 - List of public keys for allowed block publishers. For devmode
- sawtooth.gossip.time_to_live
 - Expiration time for the Gossip node communication protocol
- sawtooth.identity.allowed_keys
 - List of keys allowed to make identity transactions to the identity TP
- sawtooth.poet.enclave_module_name

- Python module name implementing the PoET enclave. Set to `sawtooth_poet_sgx.poet_enclave_sgx.poet_enclave`
- `sawtooth.poet.initial_wait_time`
 - For C Test: initial time to wait in seconds before proposing a block (e.g., 25; default is 3000)
- `sawtooth.poet.block_claim_delay`
 - For C Test: block claim delay in blocks. Set to 1 to prevent most reasonable attacks. Set to 2 or 3 if you want more aggressive protection. Default is 1.
- `sawtooth.poet.key_block_claim_limit`
 - For K Test: maximum number of blocks a validator may claim with a PoET keypair before it needs to refresh its signup information. Default is 250
- `sawtooth.poet.population_estimate_sample_size`
 - Sample size, in blocks, to compute the local mean wait time (default 50).
- `sawtooth.poet.report_public_key_pem`
 - Public key used by Validator Registry TP to verify attestation reports.
- `sawtooth.poet.target_wait_time`
 - Target time to wait in seconds before proposing a block (e.g., 5; default 20)
- `sawtooth.poet.valid_enclave_basenames`
 - Adds the enclave basename for your enclave to the blockchain for the validator registry transaction processor to use to check signup information.
- `sawtooth.poet.valid_enclave_measurements`
 - Adds the enclave measurement for your enclave to the blockchain for the validator registry transaction processor to use to check signup information.
- `sawtooth.poet.ztest_minimum_win_count`
 - For Z Test: minimum win count, to test a node is not winning too frequently.
- `sawtooth.publisher.max_batches_per_block`
 - Maximum batches allowed per block (e.g., 100)

- `sawtooth.settings.vote.approval_threshold`
 - Minimum number of votes required to accept or reject a proposal (default 1)
- `sawtooth.settings.vote.authorized_keys`
 - List of public keys for authorized voters for on-chain settings. The initial setting is in the Genesis Block, Block 0
- `sawtooth.settings.vote.proposals`
 - List of proposals to make changes to settings
- `sawtooth.swa.administrators`
 - List of public keys for authorized administrators to create, change, or delete Sabre contract and namespace registries.
- `sawtooth.validator.batch_injectors`
 - Comma-separated list of batch injectors to load. Parsed by validator at beginning of block publishing for each block
- `sawtooth.validator.block_validation_rules`
 - On-chain validation rules; enforced by the block validator
- `sawtooth.validator.max_transactions_per_block`
 - Maximum transactions allowed per block
- `sawtooth.validator.transaction_families`
 - List of permitted transaction families. If not set, all transaction families are permitted.
- `transactor`
 - Public keys of authorized signers (of any kind, batch or transaction)
- `transactor.batch_signer`
 - Public keys of authorized batch signers
- `transactor.transaction_signer`
 - Public keys of authorized transaction signers
- `transactor.transaction_signer.<transaction family name>`
 - Public keys of authorized transaction signers for a transaction processor.
- `transactor.transaction_signer.intkey`
 - Public keys of authorized intkey TF signers
- `transactor.transaction_signer.sawtooth_identity`
 - Public keys of authorized sawtooth_identity TF signers

- `transactor.transaction_signer.settings`
 - Public keys of authorized settings TF signers
- `transactor.transaction_signer.validator_registry`
 - Public keys of authorized validator_registry TF signers
- `transactor.transaction_signer.xo`
 - Public keys of authorized xo TF signers