Cli-Command

CLI Command

- The Sawtooth command-line interface (**CLI**) provides a set of commands to interact with Sawtooth services.
- Important sawtooth cli commands are

  - sawtooth - To interact with validators or validator networks.
  - sawadm - For Sawtooth administration tasks.
  - sawset - To work with settings proposals.
  - sawnet - To interact with an entire network of Sawtooth nodes.
  - poet - Initializes the Proof of Elapsed Ti me (PoET) consensus

sawtooth Command

- Interact with a Sawtooth validator, such as batches, blocks, identity, keygen, peers, settings, state, and transaction information

batch	Displays information about batches and submit new batches
block	Displays information on blocks in the current blockchain
keygen	Creates user signing keys
state	Displays information on the entries in state
transaction	Shows information on transactions in the current chain
identity	Works with optional roles, policies, and permissions
peer	Displays information about validator peers
settings	Displays on-chain settings
status	Display information related to a validator's status

Logging Into The Container

- The client container is used to run Sawtooth commands.
- Log into the client container by opening a new terminal window and running the following command.

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

* sawtooth-shell-default specifies the client container name

Sawtooth batch

The sawtooth batch subcommands display information about the Batches in the current blockchain.

sawtooth batch list

- It returns the id of each Batch, the public key of each signer, and the number of transactions in each Batch.

```
sawtooth batch list --url  http://rest-api:8008
```

Sawtooth batch

sawtooth batch show

- It returns complete information for this batch in either YAML (default) or JSON format.

```
sawtooth batch show [batch_id]  --url  http://rest-api:8008
```

Sawtooth batch

sawtooth batch status

- It returns the status of one or more batches
- Specified as a list of comma-separated Batch ids. The output is in either YAML (default) or JSON format, and includes the ids of any invalid transactions with an error message explaining why they are invalid.

```
sawtooth batch status [batch_id] --url  http://rest-api:8008
```

Sawtooth block

- The sawtooth block subcommands display information about the blocks in the current blockchain.

sawtooth block list

- The sawtooth block list subcommand queries the Sawtooth REST API for a list of blocks in the current chain.

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

Sawtooth block

sawtooth block show

- It returns complete information for this block in either YAML (default) or JSON format.
- Use the --key option to narrow the returned information to just the value of a single key, either from the batch or its header.

```
sawtooth block show [block_id] --url  http://rest-api:8008
```

Sawtooth state

sawtooth state list

- This subcommand returns the address of each entry, its size in bytes, and the byte-encoded data it contains.
- It also returns the head block for which this data is valid.

```
sawtooth state list --url http://rest-api:8008
```

Sawtooth state

sawtooth state show

- It returns the data stored at this state address and the id of the chain head for which this data is valid.
- This data is byte-encoded per the logic of the transaction family that created it, and must be decoded using that same logic.

```
sawtooth state show [state  entry address] --url http://rest-api:8008
```

Sawtooth transaction

- The sawtooth transaction subcommands display information about the transactions in the current blockchain.

sawtooth transaction list

- It returns the id of each transaction, its family and version, the size of its payload, and the data in the payload itself.

```
sawtooth transaction list --url http://rest-api:8008
```

sawtooth transaction show

- It returns complete information for this transaction in either YAML (default) or JSON format.

```
sawtooth transaction show [transaction_id]--url http://rest-api:8008
```

Sawtooth status

The sawtooth status subcommands display information related to a validator current status, including its public network endpoint and its peers.

sawtooth status show

```
sawtooth status show  --url http://rest-api:8008
```

Sawtooth keygen

sawtooth keygen

- The sawtooth keygen subcommand generates a private key file and a public key file so that users can sign Sawtooth transactions and batches.
- These files are stored in the <key-dir> directory in <key_name>.priv and <key_dir>/<key_name>.pub.

```
sawtooth keygen key_name
```

Cli Commands For Validators

Sawadm

Administration tasks such as creating the genesis batch file or validator key generation

- The sawadm subcommands create validator keys during initial configuration and help create the genesis block when initializing a validator.

sawadm genesis

- The sawadm genesis subcommand produces a file for use during the initialization of a validator.
- A network requires an initial block (known as the genesis block) whose signature will determine the blockchain id.
- This initial block is produced from a list of batches, which will be applied at genesis time.

Sawadm

sawadm keygen

- The sawadm keygen subcommand generates keys that the validator uses to sign blocks. This system-wide key must be created during Sawtooth configuration.

- Validator keys are stored in the directory /etc/sawtooth/keys/.

- By default, the public-private key files are named validator.priv and validator.pub. Use the <key-name> argument to specify a different file name.

Cli Command For Sawtooth Network

Sawnet

- The sawnet command is used to interact with an entire network of Sawtooth nodes.

sawnet compare-chains

- The sawnet compare-chains subcommand compares chains across the specified nodes.

```
sawnet compare-chains  --url http://validator:8008
```

Sawset

sawset genesis

Interact with Sawtooth network, such as comparing chains across nodes

- The sawset genesis subcommand creates a Batch of settings proposals that can be consumed by sawadm genesis and used during genesis block construction.

sawset proposal

- The Settings transaction family supports a simple voting mechanism for applying changes to on-change settings.
- The sawset proposal subcommands provide tools to view, create and vote on proposed settings.
  - **sawset proposal create**
  - **sawset proposal list**
  - **sawset proposal vote**

poet

- The poet command initializes the Proof of Elapsed Time (PoET) consensus mechanism for Sawtooth by generating enclave setup information and creating a Batch for the genesis block.

- **poet registration**

  The poet registration subcommand provides a command to work with the PoET validator registry.

- **poet registration create**

The poet registration create subcommand creates a batch to enroll a validator in the network's validator registry. It must be run from the validator host wishing to enroll.

THANK YOU