

STELLAR CLI

A R C A N E S P E L L B O O K

STELLAR
CLUE SET 3



Stellar CLI Manual (Condensed)

Use the Stellar CLI to work with Stellar accounts, contracts, and assets. This manual provides command references and examples.

Core Capabilities

- **Key management:** `stellar keys`
 - **Soroban contract interaction:** `stellar contract`
 - **Stellar Asset Contract (SAC) deployment:** `stellar contract asset`, `stellar contract id asset`
 - **Event streaming:** `stellar events`
 - **Local testnet containers:** `stellar container`
 - **XDR manipulation:** `stellar xdr`
 - **Classic transaction handling:** `stellar tx`
-

Additional Information

- **Stellar Docs:** <https://developers.stellar.org>
 - **Smart Contract Docs **:
<https://developers.stellar.org/docs/build/smart-contracts/overview>
 - **CLI Docs **: <https://developers.stellar.org/docs/tools/developer-tools/cli/stellar-cli>
-

Getting Started Example

1. Generate & Fund Identity

```
Unset
stellar keys generate alice --network testnet --fund
```

- Use the identity (`alice`) with `--source` in relevant commands.

2. Deploy a Contract (Requires `contract.wasm`)

Unset

```
stellar contract deploy --wasm ./path/to/contract.wasm --source alice --network testnet --alias my_contract
```

- Outputs Contract ID; `--alias` saves it locally.

3. Invoke a Contract Function

Pass arguments after `--`.

Unset

```
# Example: Initialize contract stellar contract invoke --id my_contract --source alice --network testnet -- initialize --admin alice --param_value 10

# Example: Call another function stellar contract invoke --id my_contract --source alice --network testnet -- process --input_data "some_data"
```

Global Options

These apply to most `stellar` commands:

Option	Description
<code>--global</code>	Use global config <code>~/.config/stellar/</code> ()
<code>--config-dir <DIR></code>	Specify config directory (default: ".")
<code>-f, --filter-logs <FILTER></code>	Filter log output (e.g., <code>RUST_LOG</code> format)
<code>-q, --quiet</code>	Suppress logs
<code>-v, --verbose</code>	Log DEBUG events
<code>--very-verbose</code>	Log DEBUG and TRACE events

Option	Description
--list	List installed plugins
--no-cache	Disable local caching for simulations/transactions

Top-Level Commands

stellar contract

Tools for Soroban smart contracts.

Subcommands: asset, alias, bindings, build, extend, deploy, fetch, id, info, init, upload, invoke, optimize, read, restore.
(Deprecated: inspect, install)

Command Details (selected examples):

stellar contract build

Builds Soroban contract WASM from a Rust Cargo project.

```
Unset
cd path/to/my_contract_crate
stellar contract build
```

- Outputs WASM to target/wasm32-unknown-unknown/release/
- Options: --package, --features, --out-dir

stellar contract extend

Extend Time-To-Live (TTL) for contract instances/data to prevent archival.

```
Unset
# Extend instance & code
```

```
stellar contract extend --id <ID_OR_ALIAS> --ledgers-to-extend 535679 --source <IDENTITY> --network testnet --durability persistent

# Extend persistent data (Symbol Key) stellar contract extend --id <ID_OR_ALIAS> --key MY_KEY --durability persistent --ledgers-to-extend 535679 --source <IDENTITY> --network testnet

# Extend persistent data (XDR Key) stellar contract extend --id <ID_OR_ALIAS> --key-xdr <BASE64_XDR> --durability persistent --ledgers-to-extend 535679 --source <IDENTITY> --network testnet
```

- Note: 535679 ledgers ≈ 30 days (max extension).

stellar events

Streams contract events from the network.

Unset

```
stellar events --network testnet --id <ID_OR_ALIAS> --start-ledger latest #  
Optional: --cursor <CURSOR>, --count 100, --topic <TOPIC_FILTER>, --output  
json
```

stellar keys

Manage local identities (keypairs).

- Add identity:

Unset

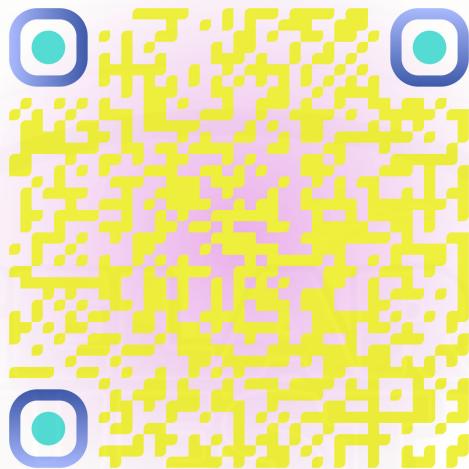
```
stellar keys add <NAME> stellar keys add  
<NAME> --public-key <PUBKEY>
```

- Generate new identity:

Unset

```
stellar keys generate <NAME> --network testnet --fund
```

```
"inclusionProof": {
  "logIndex": "73121544",
  "rootHash": "qXH9VlGWhOkfOfKCu+JCH9xX8wvyNi1x/PsMtE5tvgl=",
  "treeSize": "73121545",
  "hashes": [
    "TzbzfzielUv52sAXhHZTTbR3Wmo55gQhMWjUpPYQyn0=",
    "rC8elMMb8zwXlt6A7A3/L1KAAdHZtJKXI0ls1u115yaA=",
    "nhs9tValYWpLv3yXHANv46FAqP0CGJ8611AxK76cb/A=",
    "Dqj46Kw/c9prZgbUErGatumObyRsKYwSBKe8ZvEUfZg=",
    "H1E7AwZU3MV6mVtlkQCdwgpTqS6/bNHp2SBok1rNzxM=",
    "403yESnAOvD3ePJ0x3GOwSJ83jXJeq6F4+l88682R38=",
    "/J6n36dLPID9tb/tzE/cOC1gWdn2gnAIRtJwpzPhRN0=",
    "BnvmTaHXpvIJILEQjii7a/gh35DpyvKunDn/0V0KM0o=",
    "0YujEqJM+T9tT/ybxgkjivGDY0HjSwKNG/MEMyHh2DA=",
    "yp9AT+ooRRJm2FWNDiYQQTnoQqpPUlKr7meXncVHLfQ=",
    "6h22gZfYpQRVkz9rhZj4qY5BdYUc+f6MnadkgBgy1YI=",
    "qVmWhg0f1pqJYoBCEXskX+bB5zleSjNYmmtoDOSWOM=",
    "WEm5OgPzJpYROv+4CcriexCYyQKrLUH3hbxmlQQ+DM=",
    "7v8qPHNDLerpduaMx06eb/MwgoQwczTn/cYGKX/9wZ4="
  ],
  "checkpoint": {
    "envelope": "rekor.sigstore.dev -\n1193050959916656506\\n73121545\\nqXH9VlGWhOkfOfKCu+JCH9xX8wvyNi1x/PsMtE5tvgl=\\n\\n-\nrekor.sigstore.dev\nwNI9ajBFAiEAoSfhckTjLoxo7Dps8Ov1JAqZAihLdLagXynmEKlhSKACIH0e8qDhnzf+BKJWEjUBULD+0wrpObG/L7\nqN0adyXRm\\n"
  }
},
```



"certificate": {

 "rawBytes":

```
"MIIHKTCBq+gAwIBAgIUAAn5Pg3yCuD7/xrA9N7STH2ytoLowCgYIKoZIzj0EAwMwNzEVMBMGA1UEChMMc2lnc3R
vcmUuZGV2MR4wHAYDVQQDExVzaWdzdG9yZS1pbnRlcmlZGlhdGUwHhcNMjUwNDEwMTUyNjAwWhcNMjUwN
DEwMTUzNjAwWjAAAMFkwEwYHKOZIzj0CAQYIKoZIzj0DAQcDQgAEiYMHW2dyBH7j1VxEPvgW35u0junaz40TXkOly
LGDVmwsLza2E2GrJo/kx98U5Weg4B6HoM3rnQjKwhduIKJaaqOCBc4wggXKMA4GA1UdDwEB/wQEAvIHgDATBg
NVHSUEDDAKBggrBgEFBQcDAzAdBgNVHQ4EFgQUf9Cuw7hvww6VbjWD0BVwGNrswHwYDVR0jBBgwFoAU3
9Ppz1YkEZb5qNjpKFWixi4YZD8wcwYDVR0RAQH/BGkwZ4ZlaHR0cHM6Ly9naXRodWIuY29tL2FuYXRhbGlvY3MvU
3RlbGxhci1RdWVzdC0zLUpla3lsbC8uZ2l0aHVl3dvcmtnbG93cy9qZwt5bGwueW1sQHJZnMvaGVhZHMvZ2gtcG
FnZXwOQYKKwYBBAGDvzABAQQraHR0cHM6Ly90b2tlbi5hY3Rpb25zLmdpdGh1YnVzZXJjb250ZW50LmNvbTAS
BgorBgEEAYO/MAECBARwdXNoMDYGCisGAQQBg78wAQMEKGfhMDUxZDczOWFkNzJmZjY0OWY4YjE0MTZiMzA3
YTM1OTIiODlwMmYwKQYKKwYBBAGDvzABBAQbRGVwbG95IEpla3lsbCBzaXRlIHRvlfBhZ2VzMC8GCisGAQQBg78
wAQUEIWxFYRhbGlvY3MvU3RlbGxhci1RdWVzdC0zLUpla3lsbDAhBgorBgEEAYO/MAEGBBNyZWZzL2hLYWRzL2d
oLXBhZ2VzMdsGCisGAQQBg78wAQgELQwraHR0cHM6Ly90b2tlbi5hY3Rpb25zLmdpdGh1YnVzZXJjb250ZW50Lm
NvbTB1BgorBgEEAYO/MAEJBGcMZWh0dBzoi8vZ2l0aHVlMnVbS9hbmF0YWxpb2NzL1N0ZWxsYXItUXVlC3QtM
y1KZwt5bGwvLmdpdGh1Yi93b3JrZmxvd3MvamVreWxsLnltbEByzWZzL2hLYWRzL2doLXBhZ2VzMDgGCisGAQQB
g78wAQoEKgwoYWEwNTFkNzM5YWQ3MmZmNjQ5ZjhiMTQxNmIzMddhMzU50WU4MjAyZjAdBgorBgEEAYO/MA
ELBA8MDWdpdGh1Yi1ob3N0ZWQwRAYKKwYBBAGDvzABDAQ2DRodHRwczovL2dpdGh1Yi5jb20vYW5hdGFsaW
9jcy9TdGVsbGFyLVF1ZXN0LTMtSmVreWxsMDgGCisGAQQBg78wAQoEKgwoYWEwNTFkNzM5YWQ3MmZmNjQ5Zj
hiMTQxNmIzMddhMzU50WU4MjAyZjAjBgorBgEEAYO/MAEOBBUME3JZnMvaGVhZHMvZ2gtcFnZXMwGQYKKw
YBBAGDvzABDwQLDAk5NTY1MjU5NTAwLQYKKwYBBAGDvzABEAQfDB1odHRwczovL2dpdGh1Yi5jb20vYW5hdGF
saW9jczAXBgorBgEEAYO/MAERBAkMBzQ3NTA0MjYwdQYKKwYBBAGDvzABEgRnDGvodHRwczovL2dpdGh1Yi5jb
20vYW5hdGFsaW9jcy9TdGVsbGFyLVF1ZXN0LTMtSmVreWxsLy5naXRodWIvd29ya2Zsb3dzL2pla3lsbC55bWxAcmV
mcy9oZWFKcy9naC1wYwDlcA4BgorBgEEAYO/MAETBCoMKGFhMDUxZDczOWFkNzJmZjY0OWY4YjE0MTZiMzA3
YTM1OTIiODlwMmYwFAYKKwYBBAGDvzABFAQGDARwdXNoMGgGCisGAQQBg78wARUEwgxYaHR0cHM6Ly9naXR
odWIuY29tL2FuYXRhbGlvY3MvU3RlbGxhci1RdWVzdC0zLUpla3lsbC9hY3Rpb25zL3J1bnMvMTQzODQzMTIyMTUv
YXR0ZW1wdHMvMTAWBgorBgEEAYO/MAEWBAgMBnB1YmxpYzCbiQYKKwYBBAHWeQIEAgR7BhkAdwB1AN09MG
rGxxEyYxkeHJlnNwKiSl643jyt/4eKcoAvKe60AAABliBO2ysAAAQDAEYwRAIgXdUV214d2DfRncdyM/bQ9F1iO5yZ/O
C06OlwWQJr/vkCIE6pwh4ul6UYL7v9DjQ/cXxr68b+jIELzjCA3FOmKu2fMAoGCCqGSM49BAMDA2gAMGUCMQC+8P
o21IFinxVMDs9teXi11w220gKYvFxw1Vvoz+fqTzqO/+WvzCjWTXadOGJCAR8CMFKh1L3pKeaMxZAfnOmTJfEMrXr
W3CHI5qSE0H5fn37a4sbFE+aW4LbxtOhzBCw6tQ=="
```

}