

Sui has Mainnet, Devnet, and Testnet networks available. You can use one of the test networks, Devnet or Testnet, to experiment with the version of Sui running on that network. You can also spin up a [local Sui network](#) for local development.

The Sui Testnet and Devnet networks consist of:

- Four validator nodes operated by Mysten Labs. Clients send transactions and read requests via this endpoint: <https://fullnode.<SUI-NETWORK-VERSION>.sui.io:443> using [JSON-RPC](#).
- A public network [Sui Explorer](#) for browsing transaction history.

You can [request test SUI tokens](#) through the Sui [devnet-faucet](#) and [testnet-faucet](#) Discord channels, depending on which version of the network you use. If connected to Localnet, use cURL to request tokens from your [local faucet](#). The coins on these networks have no financial value. For obvious reasons, there is no faucet service for Mainnet.

See announcements about Sui in the [#announcements](#) Discord channel.

See the [terms of service](#) for using Sui networks.

## Tools

Sui provides the following tools to interact with Sui networks:

- [Sui command line interface \(CLI\)](#)
  - Create and manage your private keys
  - Create example NFTs
  - Call and publish Move modules
- [Sui Explorer](#) to view transactions and objects on the network

## Environment set up

First, [Install Sui](#). After you install Sui, [request SUI test tokens](#) through Discord for the network you are using: [Devnet](#) or [Testnet](#). If connected to Localnet, use cURL to request tokens from your [local faucet](#).

To check whether Sui is already installed, run the following command:

```
which sui
```

If Sui is installed, the command returns the path to the Sui binary. If Sui is not installed, it returns [sui not found](#).

See the [Sui Releases](#) page to view the changes in each Sui release.

## Configure Sui client

If you previously ran [sui genesis](#) to create a local network, it created a Sui client configuration file (client.yaml) that connects to [localhost](#) at <http://0.0.0.0:9000>. See [Connect to custom RPC endpoint](#) to update the client.yaml file.

To connect the Sui client to a network, run the following command:

```
sui client
```

If you receive the `sui-client` help output in the console, you already have a `client.yaml` file. See [Connect to custom RPC endpoint](#) to add a new environment alias or to switch the currently active network.

The first time you start Sui client without having a `client.yaml` file, the console displays the following message:

```
Config file ["<PATH-TO-FILE>/client.yaml"] doesn't exist, do you want to
connect to a Sui Full node server [y/N]?
```

Press **y** and then press **Enter**. The process then requests the RPC server URL:

```
Sui Full node server URL (Defaults to Sui Devnet if not specified) :
```

Press **Enter** to connect to Sui Devnet. To use a custom RPC server, Sui Testnet, or Sui Mainnet, enter the URL to the correct RPC endpoint and then press **Enter**.

If you enter a URL, the process prompts for an alias for the environment:

```
Environment alias for [<URL-ENTERED>] :
```

Type an alias name and press **Enter**.

```
Select key scheme to generate keypair (0 for ed25519, 1 for secp256k1, 2
for secp256r1):
```

Press **0**, **1**, or **2** to select a key scheme and then press **Enter**.

Sui returns a message similar to the following (depending on the key scheme you selected) that includes the address and 24-word recovery phrase for the address:

```
Generated new keypair for address with scheme "ed25519"
[0xb9c83a8b40d3263c9ba40d551514fbac1f8c12e98a4005a0dac072d3549c2442]
Secret Recovery Phrase : [cap wheat many line human lazy few solid bored
proud speed grocery raise erode there idea inform culture cousin shed
sniff author spare carpet]
```

## Connect to a custom RPC endpoint

If you previously used `sui genesis` with the force option (`-f` or `--force`), your `client.yaml` file already includes two RPC endpoints: `localnet` at `http://0.0.0.0:9000` and `devnet` at `https://fullnode.devnet.sui.io:443`. You can view the defined environments with the `sui client envs` command, and switch between them with the `sui client switch` command.

If you previously installed a Sui client that connected to a Sui network, or created a local network, you can modify your existing `client.yaml` file to change the configured RPC endpoint. The `sui client` commands that relate to environments read from and write to the `client.yaml` file.

To check currently available environment aliases, run the following command:

```
sui client envs
```

The command outputs the available environment aliases, with `(active)` denoting the currently active network.

```
localnet => http://0.0.0.0:9000 (active)
devnet => https://fullnode.devnet.sui.io:443
```

To add a new alias for a custom RPC endpoint, run the following command. Replace values in `<>` with values for your installation:

```
sui client new-env --alias <ALIAS> --rpc <RPC-SERVER-URL>
```

To switch the active network, run the following command:

```
sui client switch --env <ALIAS>
```

If you encounter an issue, delete the Sui configuration directory (`~/.sui/sui_config`) and reinstall the latest [Sui binaries](#).

## Validating

In the following sections, the object IDs, addresses, and authority signatures used are example values only. Sui generates unique values for each of these, so you see different values when you run the commands.

## Request test tokens

1. Join [Discord](#). If you try to join the Sui Discord channel using a newly created Discord account you may need to wait a few days for validation.
2. Get your Sui client address: `sui client active-address`

- ## Publish a Move module

```
sui client publish <your-sui-repo>/sui_programmability/examples/move_tutorial --gas-budget 30000
```

[illegible]

```

Created Objects:
  - ID: 0x0c1c0e82873b745509cecf62c341679cb5b543b866b7c8defcb38bb04089305a
, Owner: Immutable
  - ID: 0x72ba48a19cbde3aefce5a7408c0a1c15dd7656ee224adc0a6bc465a4f358a860
, Owner: Account Address (
0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be )
  - ID: 0xa8e06fa6a7e0abc7bca7df78ce7414459c034a56e6a8a08add0999bc72d3d0a9
, Owner: Account Address (
0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be )
Mutated Objects:
  - ID: 0x2750aed381dbb8cb6908cb1c0a977afe2b4fa5f6aa50f8fa146078a70fcad6de
, Owner: Account Address (
0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be )

----- Events -----
Array []
----- Object changes -----
Array [
  Object {
    "type": String("mutated"),
    "sender":
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    "owner": Object {
      "AddressOwner":
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    },
    "objectType": String("0x2::coin::Coin<0x2::sui::SUI>"),
    "objectId":
String("0x2750aed381dbb8cb6908cb1c0a977afe2b4fa5f6aa50f8fa146078a70fcad6de
"),
    "version": Number(5),
    "previousVersion": Number(4),
    "digest": String("FmzCSQC1dGHcQRZjFrqF4JnoyuVcgriRXsuNLe64FGRg"),
  },
  Object {
    "type": String("published"),
    "packageId":
String("0x0c1c0e82873b745509cecf62c341679cb5b543b866b7c8defcb38bb04089305a
"),
    "version": Number(1),
    "digest": String("BTakV197w43hoXXQZQzo3wczMyzDCyTfGt4dRHubWj1X"),
    "modules": Array [
      String("my_module"),
    ],
  },
  Object {
    "type": String("created"),
    "sender":
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    "owner": Object {
      "AddressOwner":

```

```

String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    },
    "objectType": String("0x2::package::UpgradeCap"),
    "objectId":
String("0x72ba48a19cbde3aefce5a7408c0a1c15dd7656ee224adc0a6bc465a4f358a860
"),
    "version": Number(5),
    "digest": String("FRSNipvHZMTeUGpAv14aYrt3eBeexe6TjPkjJ1ZA4phA"),
  },
  Object {
    "type": String("created"),
    "sender":
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    "owner": Object {
      "AddressOwner":
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    },
    "objectType":
String("0xc1c0e82873b745509cecf62c341679cb5b543b866b7c8defcb38bb04089305a:
:my_module::Forge"),
    "objectId":
String("0xa8e06fa6a7e0abc7bca7df78ce7414459c034a56e6a8a08add0999bc72d3d0a9
"),
    "version": Number(5),
    "digest": String("CSWeoFXAf7kMrSQqT6fivWThHgvHbKXzwyjst44vas8b"),
  },
]
----- Balance changes -----
Array [
  Object {
    "owner": Object {
      "AddressOwner":
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    },
    "coinType": String("0x2::sui::SUI"),
    "amount": String("-1121"),
  },
]

```

The package publish operation creates several objects:

- A package object (with ID `0x0c1c...305a` in the example output).
- A `Forge` object (with ID `0xa8e0...d0a9` in the example output) as a result of running a [module initializer](#) for the one (and only) module of this package.
- An `UpgradeCap` object (with ID `0x72ba...a860` in the example output) for use in future package upgrades.

You can check the details of each object using the `sui client object <OBJECT-ID>` command or by using the Sui Explorer.

When you publish a package, the IDs for the objects the compiler creates are different than the ones displayed in this example. The remainder of this topic uses `<PACKAGE-ID>` and `<FORGE-ID>` to represent the IDs for the created objects (except for console output examples). You must replace these with the values for your package.

## Make a Move call

This section describes how to call into functions defined in the package published in the previous section. Use the `<PACKAGE-ID>` and `<FORGE-ID>` values from your package to create swords and transfer them to other players.

To demonstrate this, `<PLAYER-ADDRESS>` represents the address of the player to receive a sword. You can use an address of someone you know, or create another address for testing with the following Sui Client CLI command:

```
sui client new-address ed25519
```

The command returns the following message and a 24-word recovery phrase for the address:

```
Created new keypair for address with scheme ED25519:
[0xa01cd0c520f12a1e9d57bf3cc6ea0f8cf93e81e9fe46f7b4916c310a809dfddd]
Secret Recovery Phrase : [sunny tip element salad frequent february amount
notice chair kite race push noise ketchup that same cannon bench mirror
please dinosaur indicate violin sunset]
```

To create a sword and transfer it to another player, use the following command to call the `sword_create` function in the `my_module` module of the package you previously published.

You must use the same format for the command and function parameters as the example shown.

```
sui client call --function sword_create --module my_module --package
0x<PACKAGE-ID> --args \"0x<FORGE-ID>\" 42 7 \"0x<PLAYER-ADDRESS>\" --gas-
budget 30000
```

The response resembles the following:

```
----- Transaction Digest -----
5JqthvBVWgCaDy77G1XSjZjw6eysx6WVrcLYrKPDtmK
----- Transaction Data -----
Transaction Signature:
[Signature(Ed25519SuiSignature(Ed25519SuiSignature([0, 4, 117, 62, 101,
51, 245, 143, 229, 126, 215, 58, 173, 66, 58, 216, 191, 47, 107, 5, 20,
```

```
78, 67, 167, 206, 93, 99, 163, 224, 152, 16, 12, 131, 121, 160, 87, 201,
27, 26, 62, 228, 185, 73, 28, 228, 98, 8, 253, 199, 218, 215, 165, 235,
231, 127, 220, 20, 189, 72, 19, 164, 201, 191, 213, 12, 198, 105, 207,
220, 225, 156, 27, 143, 143, 103, 80, 186, 158, 114, 177, 254, 3, 59, 146,
37, 220, 209, 19, 199, 189, 131, 122, 244, 243, 74, 19, 121]]))]]
```

Transaction Kind : Programmable

```
Inputs: [Object(ImmOrOwnedObject { object_id:
0xa8e06fa6a7e0abc7bca7df78ce7414459c034a56e6a8a08add0999bc72d3d0a9,
version: SequenceNumber(5), digest:
o#CSWeoFXAf7kMrSQqT6fivWThHgvHbKXzwyjst44vas8b }), Pure(SuiPureValue {
value_type: Some(U64), value: "42" }), Pure(SuiPureValue { value_type:
Some(U64), value: "7" }), Pure(SuiPureValue { value_type: Some(Address),
value:
"0x72ba48a19cbde3aefce5a7408c0a1c15dd7656ee224adc0a6bc465a4f358a860" })]
```

Commands: [

```
MoveCall(0x0c1c0e82873b745509cecf62c341679cb5b543b866b7c8defcb38bb04089305
a::my_module::sword_create(,Input(0),Input(1),Input(2)Input(3))),
]
```

Sender: 0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be

Gas Payment: Object ID:

0x2750aed381dbb8cb6908cb1c0a977afe2b4fa5f6aa50f8fa146078a70fcad6de,

version: 0x5, digest: FmzCSQC1dGHcQRZjFrqF4JnoyuVcgriRXsuNLe64FGRg

Gas Owner:

0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be

Gas Price: 1

Gas Budget: 30000

----- Transaction Effects -----

Status : Success

Created Objects:

- ID: 0x896ecfe58c9c6326284be0ce554a12b0d14158f6f7b4e3d5137ebe77488ccba6

, Owner: Account Address (

0x72ba48a19cbde3aefce5a7408c0a1c15dd7656ee224adc0a6bc465a4f358a860 )

Mutated Objects:

- ID: 0x2750aed381dbb8cb6908cb1c0a977afe2b4fa5f6aa50f8fa146078a70fcad6de

, Owner: Account Address (

0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be )

- ID: 0xa8e06fa6a7e0abc7bca7df78ce7414459c034a56e6a8a08add0999bc72d3d0a9

, Owner: Account Address (

0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be )

----- Events -----

Array []

----- Object changes -----

Array [

Object {

"type": String("mutated"),

"sender":

String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be"),

"owner": Object {

"AddressOwner":



```

String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    },
    "objectType": String("0x2::coin::Coin<0x2::sui::SUI>"),
    "objectId":
String("0x2750aed381dbb8cb6908cb1c0a977afe2b4fa5f6aa50f8fa146078a70fcad6de
"),
    "version": Number(6),
    "previousVersion": Number(5),
    "digest": String("DTjcskKvVbd6xtuFac8iimQrD3msfj2MoZ1RssfqlPNC"),
  },
  Object {
    "type": String("mutated"),
    "sender":
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    "owner": Object {
      "AddressOwner":
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    },
    "objectType":
String("0xc1c0e82873b745509cecf62c341679cb5b543b866b7c8defcb38bb04089305a:
:my_module::Forge"),
    "objectId":
String("0xa8e06fa6a7e0abc7bca7df78ce7414459c034a56e6a8a08add0999bc72d3d0a9
"),
    "version": Number(6),
    "previousVersion": Number(5),
    "digest": String("GXCT1KohW18DkGFTcgMFoWioMumnNmMGdR3m6BjMq8qw"),
  },
  Object {
    "type": String("created"),
    "sender":
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be
"),
    "owner": Object {
      "AddressOwner":
String("0x72ba48a19cbde3aefce5a7408c0a1c15dd7656ee224adc0a6bc465a4f358a860
"),
    },
    "objectType":
String("0xc1c0e82873b745509cecf62c341679cb5b543b866b7c8defcb38bb04089305a:
:my_module::Sword"),
    "objectId":
String("0x896ecfe58c9c6326284be0ce554a12b0d14158f6f7b4e3d5137ebe77488ccba6
"),
    "version": Number(6),
    "digest": String("7Ti1xqneQ1Pg2mEv8rW7Fd28gjH1jCTtSomzJYbQHg4J"),
  },
]
----- Balance changes -----
Array [
  Object {

```

```
    "owner": Object {  
      "AddressOwner":  
String("0xb0f74e94014954b5c8a3abccbcfde2bf014c906a3fec1bb9736928f415a1f0be  
"),  
    },  
    "coinType": String("0x2::sui::SUI"),  
    "amount": String("-1018"),  
  },  
]
```

Go to the Sui Explorer to observe a newly created object. You should see a sword object created with **Magic** property of **42** and **Strength** property of **7** and transferred to the new owner.

 **Object**

**0x3265ebb34fc92ab905ec312a03044c03afaa4375dc83c44dee3c5246a8b67163** 

**Description**

Type	0xafb60ea71302446ba4bf3a1e7b0c20a644db634fd2f0e32aa4ca9354dcfa4f8::my_module::Sword
Object ID	0x3265ebb34fc92ab905ec312a03044c03afaa4375dc83c44dee3c5246a8b67163
Last Transaction ID	DCHyMBpirumN12qFTZLhH8Emb8UTF6c9SNQ6uLwc5zXm
Version	3
Owner	0xa01c...fddd

**Properties**

magic	42
strength	7

#### Explorer view of example sword object

To see your object in the current [Sui Explorer](#), paste the object ID of the created object you observed in your own command output in the search field and press **Enter**. If your Sui Explorer doesn't find your object, make sure it's pointing to the right network.

Related topics:

- [Create Smart Contracts with Move](#)
- [Programming with Objects](#)