



Usage instructions:

- Launch the product via 1-click. **Please wait until** the instance passes all status checks and is running. You can connect using your Amazon private key and 'ubuntu' login via your SSH client.
- All the dependencies have been installed. To get started with Solana Development please view documentation: <https://solana.com/docs/intro/dev>

The following are some helpful Start Commands:

Verify the Solana Environment Setup

1. **Check Solana Version:** Run the command to verify the Solana CLI is available and versioned correctly.

solana --version

2. **Check Rust Compiler:**

rustc --version

```
ubuntu@ip-172-31-30-4:~$ solana --version
solana-cli 2.0.15 (src:7feb24da; feat:4052914393, client:Agave)
ubuntu@ip-172-31-30-4:~$ rustc --version
rustc 1.82.0 (f6e511eec 2024-10-15)
```

Configure Solana CLI for Local Development

1. **Verify the Configuration:**

solana config get

*Ensure that the RPC URL is **http://127.0.0.1:8899***

```
ubuntu@ip-172-31-30-4:~$ solana config get
Config File: /home/ubuntu/.config/solana/cli/config.yml
RPC URL: http://127.0.0.1:8899
WebSocket URL: ws://127.0.0.1:8900/ (computed)
Keypair Path: /home/ubuntu/.config/solana/id.json
Commitment: confirmed
```

Request Test SOL

1. Airdrop SOL for Testing

solana airdrop 10

This command provides test SOL, which can be used to interact with the local validator.

Develop and Deploy Your First Program with Anchor

1. Log into program directory:

cd my-solana-app/my_program

2. Check the Status of the Service:

sudo systemctl status solana-test-validator

```
ubuntu@ip-172-31-30-4:~/my-solana-app/my_program$ sudo systemctl status solana-test-validator
● solana-test-validator.service - Solana Test Validator
   Loaded: loaded (/etc/systemd/system/solana-test-validator.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-11-14 19:40:06 UTC; 10min ago
     Main PID: 621 (solana-test-val)
        Tasks: 157 (limit: 19170)
      Memory: 2.4G (peak: 2.4G)
         CPU: 2min 37.101s
       CGroup: /system.slice/solana-test-validator.service
              └─621 /home/ubuntu/.local/share/solana/install/active_release/bin/solana-test-validator
```

3. Build the Program:

anchor build

This uploads your program to the local test validator.

```
ubuntu@ip-172-31-30-4:~/my-solana-app/my_program$ anchor build
Finished release [optimized] target(s) in 5.24s
Finished `test` profile [unoptimized + debuginfo] target(s) in 16.32s
Running unittests src/lib.rs (/home/ubuntu/my-solana-app/my_program/target/debug/deps/my_program-07715c62349989f1)
```

Interact with Your Program

1. Create a client script in JavaScript or Rust to interact with your program. Here's an example in JavaScript:

2. Navigate to the Tests Directory:

cd my-solana-app/my_program/tests

3. Edit Client Script (if necessary)

sudo nano client.js

- Replace "**YOUR_PROGRAM_ID**" with your actual program ID from the Anchor deployment output.
- Add additional functionality in the script based on the specific methods in your Solana program.

4. Run the Script:

node client.js

Additional Resources

1. **Solana Scaffold:** To quickly get started with a front-end for your application, you can generate a customizable Solana scaffold by typing the following into your CLI:

npx create-solana-dapp <project-name>

- This will create a new project with all the necessary files and basic configuration to get started building on Solana. The scaffold will include both an example frontend and an onchain program template (if you selected one). You can read the create-solana-dapp docs to learn more.
- <https://github.com/solana-developers/create-solana-dapp?tab=readme-ov-file#create-solana-dapp>

2. **Online IDE:** For web-based development, consider using **Solana Playground**.

See: <https://beta.solpg.io/>

Getting Support

For questions and support, the best place to start is **Solana StackExchange**. Search for your question first; if it's not there, create a new question with as much detail as possible. Remember to use text instead of screenshots for error messages, as it makes it easier for others with the same problem to find your question. See:

<https://solana.stackexchange.com/>

<https://solana.com/docs/intro/quick-start>

AWS Data

- Data Encryption Configuration: This solution does not encrypt data within the running instance.
- User Credentials are stored: /root/.ssh/authorized_keys & /home/ubuntu/.ssh/authorized_keys
- Monitor the health:
 - Navigate to your Amazon EC2 console and verify that you're in the correct region.
 - Choose Instance and select your launched instance.
 - Select the server to display your metadata page and choose the Status checks tab at the bottom of the page to review if your status checks passed or failed.

Extra Information: (Optional)

Allocate Elastic IP

To ensure that your instance **keeps its IP during restarts** that might happen, configure an Elastic IP. From the EC2 console:

1. Select ELASTIC IPs.
2. Click on the ALLOCATE ELASTIC IP ADDRESS.
3. Select the default (Amazon pool of IPv4 addresses) and click on ALLOCATE.
4. From the ACTIONS pull down, select ASSOCIATE ELASTIC IP ADDRESS.
5. In the box that comes up, note down the Elastic IP Address, which will be needed when you configure your DNS.
6. In the search box under INSTANCE, click and find your INSTANCE ID and then click ASSOCIATE.
7. Your instance now has an elastic IP associated with it.
8. For additional help: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>