

Solana Anchor Smart Contract Workbench with VS Code by Optick

Customer Instructions for AWS Marketplace Users

Important first use note****

The first time you build or validate the included Anchor starter project, Rust and Anchor dependencies may take 5 to 10 minutes to compile.

Wait until the command completes before moving to the next step. Later builds are usually much faster because dependencies are cached.

Item	Value
Product	Solana Anchor Smart Contract Workbench with VS Code by Optick
Operating system	Ubuntu 24.04 LTS
Linux user	ubuntu
Public web port	TCP 80
SSH port	TCP 22
Browser VS Code path	http://PUBLIC_IP/code/
Landing page	http://PUBLIC_IP/
Local Solana RPC	http://127.0.0.1:8899, blocked from public access by firewall
Included starter project	/opt/optick-solana-workbench/projects/optick_counter

1. What this server does

This server is a ready to use Solana and Anchor smart contract development workbench. It is designed to help developers launch an AWS instance and begin working without manually installing Rust, Solana CLI, Anchor, Node.js, Yarn, Surfpool, a browser based VS Code environment, or a local Solana validator.

- **Browser based VS Code:** Open a VS Code style development environment from your browser.
- **Solana and Anchor toolchain:** Build Solana programs using Rust, Solana CLI, Anchor CLI, Node.js, and Yarn.
- **Managed local validator:** Use a local Solana validator for development and testing on the instance.
- **Starter project:** Inspect, build, and test an included Anchor counter starter project.
- **First boot automation:** Each new instance generates its own browser VS Code password, local development keypair, public IP instructions, and first login file.
- **Firewall protection:** Public access is intended for SSH and HTTP only. Local validator ports are blocked from public access.

Local development warning

The local Solana keypair and localnet SOL generated on this server are for local development and testing only. Do not treat the generated local keypair as a production wallet. Do not send real assets to it.

2. Connect to the instance

After launching the AMI from AWS Marketplace, connect by SSH with your Amazon private key and the **ubuntu** user.

```
ssh -i your-key.pem ubuntu@PUBLIC_IP
```

Replace PUBLIC_IP with the public IPv4 address shown in the EC2 console.

Security group requirements

Open TCP 22 for SSH from your IP address.

Open TCP 80 for the landing page and browser VS Code proxy.

Do not open TCP 8080, 8899, 8900, or 9900 to the public internet. The AMI firewall blocks the local validator ports, but the AWS security group should stay restrictive too.

3. First login commands

After SSH login, read the first login instructions generated specifically for your instance.

```
cat /home/ubuntu/FIRST_LOGIN.txt
```

Show the application URLs:

```
optick-solana-url
```

Show the browser VS Code password:

```
optick-solana-password
```

Check the full application status:

optick-solana-status

4. Open browser VS Code

1. Open the landing page in your browser: `http://PUBLIC_IP/`
2. Click Open Browser VS Code or go directly to `http://PUBLIC_IP/code/`
3. When prompted, enter the password shown by `optick-solana-password`.
4. Open the workspace under `/opt/optick-solana-workbench` or go to the included project under `/opt/optick-solana-workbench/projects/optick_counter`.

If the page does not appear immediately

Wait one minute and refresh the browser page. First boot may still be finishing service startup.

5. Validate the included Solana and Anchor environment

Use the status helpers to confirm the local validator and developer tools are working.

optick-solana-status

optick-solana-localnet-status

Compile time note

The first full Anchor and Rust validation can take 5 to 10 minutes because Rust dependencies compile on first use.

Do not stop the command early. Wait until you see that the starter project validation completed successfully.

Run the included starter project validation:

optick-solana-test-starter

A successful validation should show:

- Local validator RPC is healthy.
- Anchor build completes successfully.
- Rust unit tests pass.
- Solana CLI localnet check returns a cluster version and a localnet SOL balance.

6. Use the included Anchor starter project

The included starter project is located here:

cd /opt/optick-solana-workbench/projects/optick_counter

Build it manually:

anchor build

Run Rust tests manually:

cargo test

The helper below runs the recommended stable validation path for this AMI:

optick-solana-test-starter

7. Use the managed local Solana validator

The AMI includes a managed local Solana validator for development and testing. The Solana CLI is configured to use localnet at <http://127.0.0.1:8899>.

Command	Purpose
<code>optick-solana-localnet-status</code>	Show validator health, RPC status, balance, and recent logs.
<code>optick-solana-localnet-start</code>	Start the local validator service.
<code>optick-solana-localnet-stop</code>	Stop the local validator service.
<code>optick-solana-localnet-restart</code>	Restart the local validator service.
<code>optick-solana-localnet-reset</code>	Reset the local ledger and start a clean localnet.
<code>optick-solana-localnet-logs</code>	Show local validator logs.

8. Create a new Anchor project

Use the projects directory for your own work.

```
cd /opt/optick-solana-workbench/projects
anchor init my_solana_app --no-git
cd my_solana_app
anchor build
```

You can then open the project in browser VS Code and begin editing the program, tests, and Anchor configuration files.

First build wait time

A newly created Anchor project may also take 5 to 10 minutes to compile the first time, depending on instance size and dependency cache state.

Subsequent builds are usually faster.

9. Helper command reference

Command	Purpose
<code>optick-solana-url</code>	Show landing page URL, browser VS Code URL, local validator note, and SSH reminder.
<code>optick-solana-password</code>	Show the generated browser VS Code password.
<code>optick-solana-status</code>	Check Nginx, code server, local validator, local HTTP status, toolchain versions, Solana config, and starter project location.
<code>optick-solana-projects</code>	List the Solana workbench project directory and starter project files.
<code>optick-solana-verify</code>	Show toolchain versions and Solana configuration.
<code>optick-solana-readme</code>	Open the server README file.
<code>optick-solana-test-starter</code>	Build and test the included Anchor starter project using the stable validation path.
<code>optick-solana-localnet-status</code>	Check local validator health and balance.
<code>optick-solana-localnet-reset</code>	Reset local validator ledger for a clean localnet.

10. Security notes

- The browser VS Code password is generated separately for each new instance.
- The Solana local development keypair is generated separately for each new instance.
- The local development keypair is not intended for production wallets or real assets.
- The local validator RPC runs on the instance and is intended for local development only.
- The AMI firewall denies inbound public access to ports 8899, 8900, and 9900.
- Use AWS security groups to restrict SSH access to trusted IP addresses.

11. Troubleshooting

Issue	What to do
Landing page does not load	Confirm the EC2 security group allows TCP 80. Wait one minute after launch and refresh.
Browser VS Code login does not appear	Run <code>optick-solana-status</code> . Confirm code server is active. Wait one minute and refresh <code>/code/</code> .
I do not know the VS Code password	Run <code>optick-solana-password</code> after SSH login.
Anchor build seems slow	The first Anchor and Rust build may take 5 to 10 minutes. Wait for completion.
Local validator balance looks wrong	Run <code>optick-solana-localnet-status</code> . If needed, run <code>optick-solana-localnet-reset</code> .
I want a clean localnet	Run <code>optick-solana-localnet-reset</code> .

12. Learning resources

The following official resources can help you learn more about the tools included in this AMI.

- Solana Installation and CLI docs: <https://solana.com/docs/intro/installation>
- Solana Surfpool CLI basics: <https://solana.com/docs/intro/installation/surfpool-cli-basics>
- Anchor Quickstart: <https://www.anchor-lang.com/docs/quickstart>
- Anchor CLI Reference: <https://www.anchor-lang.com/docs/references/cli>
- code-server Install Docs: <https://coder.com/docs/code-server/install>
- code-server FAQ: <https://coder.com/docs/code-server/FAQ>
- Surfpool Documentation: <https://docs.surfpool.run/>

13. Recommended AWS settings

Use case	Suggested instance	Notes
Light evaluation	t3.large	Usable for initial testing and small examples.
Comfortable development	t3.xlarge	Recommended for smoother Rust and Anchor compilation.
Heavier development	m6i.xlarge or larger	Use more CPU and memory for frequent builds or multiple users.

Recommended root volume: 50 GB gp3 or larger.

14. Quick start checklist

1. Launch the instance with public IP enabled.
2. Open TCP 22 and TCP 80 in the security group.
3. SSH as ubuntu using your Amazon private key.
4. Run `cat /home/ubuntu/FIRST_LOGIN.txt`.
5. Run `optick-solana-password` and save the browser VS Code password securely.
6. Open `http://PUBLIC_IP/` and `http://PUBLIC_IP/code/`.
7. Run `optick-solana-status`.
8. Run `optick-solana-localnet-status`.
9. Run `optick-solana-test-starter` and allow 5 to 10 minutes for the first build and tests.