

Setting Up a CLI Wallet: A Step-by-Step Guide

Introduction

This guide is designed for users looking to set up a Command Line Interface (CLI) wallet for blockchain operations. We'll use a fictitious user, Alex, as an example. Alex wants to participate in the R-Squared Labs blockchain and needs to set up a CLI wallet to manage their assets securely.

Requirements

- A computer with internet access.
- Basic familiarity with command-line operations.

Step 1: Installing Docker

1. Download Docker: Alex starts by downloading Docker Desktop from the official Docker website (<https://www.docker.com/products/docker-desktop>) suitable for their operating system (Windows/MacOS).
2. Install Docker: They follow the installation instructions, ensuring Docker is set up correctly on their system.

Step 2: Downloading the CLI Wallet Repository

1. Clone the Repository: Alex navigates to the GitHub page of the R-Squared Labs CLI wallet and clones the repository to their local machine using Git or downloads the ZIP file directly.
 - Command: `git clone https://github.com/R-Squared-Project/R2-Wallet-main.git`
2. Navigate to the Directory: They open a command prompt or terminal and navigate to the directory where they cloned or unzipped the wallet files.
 - Command: `cd path/to/R2-Wallet-main`

Step 3: Starting the CLI Wallet with Docker

1. Launch Docker: Alex ensures Docker Desktop is running.
2. Initialize the Docker Environment: They use Docker Compose to set up and start the CLI wallet environment.
 - Command: `docker-compose up -d`
3. Verify the Container is Running: Alex checks that the Docker container is up and running.
 - Command: `docker-compose ps`

Step 4: Accessing the CLI Wallet

1. Access the Wallet's Bash Shell: Alex accesses the bash shell of the Docker container to interact with the CLI wallet.
 - Command: `docker-compose exec r2_wallet bash`
2. Start the CLI Wallet: Inside the bash shell, they start the CLI wallet application.
 - Command: `/usr/local/bin/cli_wallet`

Step 5: Setting Up the Wallet

1. Set a Password: Alex initializes their new wallet by setting a password.
 - Command: ``set_password <strong_password>``
2. Unlock the Wallet: They unlock the wallet using the password they just set.
 - Command: ``unlock <strong_password>``

Step 6: Generating Cryptographic Keys

1. Generate Keys: Alex uses the CLI wallet to generate a new set of cryptographic keys (brain key, private key, and public key).

- Command: ``suggest_brain_key``

Per the community guide, this will output similar to:

```
{  
  "brain_priv_key": "EXAMPLE PRIVATE KEY",  
  "wif_priv_key": "5J...",  
  "pub_key": "RQR..."  
}
```

2. Secure Key Storage: They securely store the generated keys. The brain key is written down on paper and stored in a safe place, while the private and public keys are saved in a reputable password manager.

Conclusion

At this point, Alex has successfully set up their CLI wallet and is ready to participate in blockchain activities. They can now use their wallet to manage assets, participate in governance, or set up as a witness/node on the R-Squared Labs blockchain.

Security Tips

- Always keep your private keys secure and never share them.
- Use a strong, unique password for your wallet and password manager.
- Regularly back up your wallet data and keys.