viper-binaries

Private

main 2 Branches 0 Tags Go to file t Go to file Code

vishruthsk commit 6d88f86 · last week 23 Commits

chains&geoZones	Updated language in Chains s...	2 weeks ago
README.md	Update README.md	last week
viper_linux_amd64	commit	last week
viper_linux_arm64	commit	last week
viper_macOS_amd64	commit	last week
viper_macOS_arm64	commit	last week
viper_win_amd64.exe	commit	last week

About

This repo consists of binary files to run a viper node

vipernet.xyz/

Readme
Activity
Custom properties
22 stars
31 watching
0 forks

Releases

No releases published

No packages published

Contributors 2

vishruthsk vishruthsk.eth

ThanhTuan1695

README

Viper Node Setup Guide

Welcome to the Viper Node Setup Guide! This comprehensive walkthrough will assist you in configuring your server to run a Viper node, enabling you to participate in the network.

Hardware Requirements

```
4 CPUs (or vCPUs)
8 GB RAM
100 GB Disk
```

Prerequisites

- Configure your DNS if you have a domain.

Login with SSH

Replace `mynode.vipernet.xyz` with your own DNS name or IP address

```
ssh root@mynode.vipernet.xyz
```

Set Hostname

1. Open the `/etc/hostname` file with the `nano` editor:

```
nano /etc/hostname
```

2. Change the localhost value to your Hostname, e.g., mynode.vipernet.xyz
3. Save the file and exit the editor (press `Ctrl+O`, `Enter`, `Ctrl+X`).
4. Reboot the server:

```
reboot
```

Create New User

1. Create a new user named `viper` and add it to the `sudo` group:

```
useradd -m -g sudo -s /bin/bash viper && passwd viper
```

2. Switch to the new user:

```
su - viper
```

Install Dependencies

```
sudo apt update
```

```
sudo apt dist-upgrade -y
```

```
sudo apt-get install git build-essential curl file nginx certbot python3-certbot-nginx jq -y
```

Viper Setup

1. Create a directory for the Vipernet network:

```
sudo mkdir -p viper-network
```

```
cd viper-network
```

2. Clone the Vipernet binaries repository:

```
sudo git clone https://github.com/vipernet-xyz/viper-binaries
```

```
cd viper-binaries
```

3. Copy the binary to `/usr/local/bin/viper` (replace `$YOUR_ARCH_TYPE_BINARY` with the actual file name, e.g., `viper_linux_amd64`):

```
sudo cp $YOUR_ARCH_TYPE_BINARY /usr/local/bin/viper
```

Viper Configuration

1. Create a new account:

```
viper wallet create-account
```

2. Create a validator (replace `address` with the address obtained from the previous command):

```
viper servicers create-validator address
```

3. Fund your account by requesting test tokens from the `#🤑|req-tokens` channel at Viper Network Discord server and wait till the tx gets included in the block (It may take a couple of mintues for the amount to reflect)

4. Adding Persistent Peers

```
echo $(viper util print-configs) | jq '.tendermint_config.P2P.PersistentPeers = "859674aa64c0ee20ebce8a50e6939069875c
```

Check if the persistent peers are included in the config under P2P sec before proceeding, if not manually add them

```
cat ~/.viper/config/configuration.json
```

5. Generate chains (enter the `chain ID` and the `URL` of the non-native chain node when prompted):

```
viper util gen-chains
```

Enter the ID of the Viper network:

```
0001
```

Enter the URL of the network identifier.

```
http://127.0.0.1:8082/
```

Enter `y` if you want to add non-native chains `(Enter "n" for now)`

For Non-Native Chains:

- When prompted, enter the chain ID. This is a unique identifier for the blockchain network you want to service through the Viper node. For example, if you want to service the Ethereum network, you would enter the chain ID of Etherereum.
- Next, when prompted, enter the URL of the non-native chain node you wish to service through the Viper node. This should be the RPC endpoint of the blockchain node you want to connect to.
- Repeat steps 2 and 3 for each additional non-native chain you want to service through the Viper node.
- Once you have entered all the required chain IDs and URLs, the command will generate the necessary configuration files for those chains within the Viper node.

6. Generate the geozone (enter the `geo ID` when prompted):

```
viper util gen-geozone
```

7. Change to the `~/.viper` directory:

```
cd ~/.viper
```

8. Download state

```
sudo git clone https://github.com/vishruthsk/data.git data
```

9. Change Ownership

```
sudo chown -R viper ~/.viper/data
```

10. Change to the `config` directory:

```
cd config
```

11. Download the genesis file:

```
wget https://raw.githubusercontent.com/vipernet-xyz/genesis/main/testnet/genesis.json genesis.json
```

Increase ulimit

```
ulimit -Sn 16384
```

Create Systemd Service

1. Create a systemd service file for the Vipernet node:

```
sudo nano /etc/systemd/system/viper.service
```

2. Copy and paste the following content into the file:

```
[Unit]
Description=viper service
After=network.target
Wants=network-online.target systemd-networkd-wait-online.service

[Service]
User=viper
Group=sudo
ExecStart=/usr/local/bin/viper network start
ExecStop=/usr/local/bin/viper network stop

[Install]
WantedBy=default.target
```

3. Save the file and exit the editor (press `Ctrl+O`, `Enter`, `Ctrl+X`).
4. Reload the systemd daemon:

```
sudo systemctl daemon-reload
```

5. Enable and start the `viper.service` service:

```
sudo systemctl enable viper.service
```

```
sudo systemctl start viper.service
```

SSL Configuration

1. Obtain an SSL certificate using Certbot (replace `$HOSTNAME` with your actual hostname):

```
sudo certbot --nginx --domain $HOSTNAME --register-unsafely-without-email --no-redirect --agree-tos
```

Nginx Configuration

1. Open the Nginx configuration file:

```
sudo nano /etc/nginx/sites-available/viper
```

2. Copy and paste the following content into the file (replace `$HOSTNAME` with your actual hostname):

```
server {
    add_header Access-Control-Allow-Origin "*";
    listen 80 ;
    listen [::]:80 ;
    listen 8081 ssl;
    listen [::]:8081 ssl;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name $HOSTNAME;
```

```
    location / {
        try_files $uri $uri/ =404;
    }

    listen [::]:443 ssl ipv6only=on;
    listen 443 ssl;
    ssl_certificate /etc/letsencrypt/live/$HOSTNAME/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/$HOSTNAME/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;
    access_log /var/log/nginx/reverse-access.log;
    error_log /var/log/nginx/reverse-error.log;

    location ~* ^/v1/client/(dispatch|relay|sim|trigger) {
        proxy_pass http://127.0.0.1:8082;
        add_header Access-Control-Allow-Methods "POST, OPTIONS";
        allow all;
    }

    location = /v1 {
        add_header Access-Control-Allow-Methods "GET";
        proxy_pass http://127.0.0.1:8082;
        allow all;
    }

    location = /v1/query/height {
        add_header Access-Control-Allow-Methods "GET";
        proxy_pass http://127.0.0.1:8082;
        allow all;
    }
}
```

3. Save the file and exit the editor (press `Ctrl+O`, `Enter`, `Ctrl+X`).
4. Stop the Nginx service:

```
sudo systemctl stop nginx
```

5. Remove the default Nginx configuration:

```
sudo rm /etc/nginx/sites-enabled/default
```

6. Create a symbolic link to the new Vipernet configuration:

```
sudo ln -s /etc/nginx/sites-available/viper /etc/nginx/sites-enabled/viper
```

7. Start the Nginx service:

```
sudo systemctl start nginx
```

UFW Configuration

1. Enable UFW:

```
sudo ufw enable
```

2. Deny incoming traffic by default:

```
sudo ufw default deny
```

3. Allow SSH:

```
sudo ufw allow ssh
```

4. Allow HTTP:

```
sudo ufw allow 80
```

5. Allow HTTPS:

```
sudo ufw allow 443
```

6. Allow Vipernet-specific port (8081):

```
sudo ufw allow 8081
```

7. Allow Vipernet-specific port (26656):

```
sudo ufw allow 26656
```

Stake

Stake your validator (replace `addr`, `amount`, `chainIDs` (comma seperated; for, e.g., 0001,0002,0003) and `https://hostname:443` with your own values). When staking, you need to specify the `amount` in the smallest unit of VIPR, which is uVIPR. 1 VIPR is equal to 1,000,000 uVIPR. The minimum stake amount required is 10,000 VIPR, but it's recommended to stake around 11,000 VIPR.

```
viper servicers stake self addr amount chainIDs geo_ID https://hostname:443 testnet
```

Note: Replace `hostname` with your actual hostname

Useful Command

Check Node Status

```
curl http://127.0.0.1:26657/status
```

Query Your Validator

```
viper servicers query servicer your_wallet_address
```

Note: Replace `your_wallet_address` with the address that was used to create the validator

Wallet Command

Check Balance

```
viper wallet query account-balance your_wallet_address
```

Backup Wallet

```
viper wallet export-encrypted

# or

viper wallet export-raw
```

Recovery Wallet

```
viper wallet import-encrypted

# or

viper wallet import-raw
```

Fetch Wallet Info

```
viper wallet fetch-account your_wallet_address
```

List all account

```
viper wallet list-accounts
```

Change wallet password

```
viper wallet change-pass your_wallet_address --pwd-new input-new-passwd --pwd-old input-old-passwd
```

`input-new-passwd`: replace with your new password for the wallet

`input-old-passwd`: replace with current wallet password

Note: Replace `your_wallet_address` with the wallet that was used to create the validator

Check logs of the node

```
sudo journalctl -u viper -f -o cat
```

Restart the node