**How to start Satori Neuron docker container with wireguard private network.**

## **Overview**

WireGuard is a fast, modern, and secure VPN (Virtual Private Network) protocol. To connect a client to the WireGuard server, users need to:

1. Add their public keys to the WireGuard server.
2. Download the generated configuration file from the server.
3. Import this configuration into their WireGuard client to establish the connection.

To automate the user addition and configuration file generation process, we’ve built a Flask-based web application that:

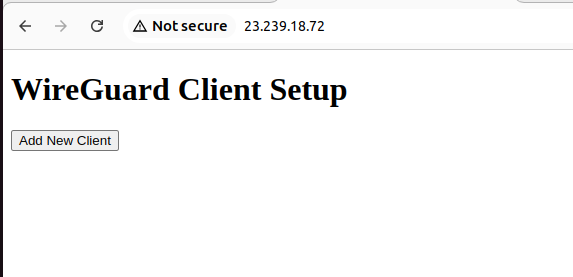
* Adds users to the WireGuard server.
* Generates a configuration file that the user can use to connect to the VPN network.
* Provides an easy-to-use web interface for users to download the configuration file.

## **System Architecture**

1. WireGuard Server: This is the central server running WireGuard, which handles the VPN connections.
2. Flask Web Application: The Flask application runs on a web server http://23.239.18.72 which allows users to register, and automatically adds them to the WireGuard server.
3. WireGuard Client: Users install the WireGuard client on their device, which they use to connect to the private network using the configuration file provided.

## **Key Steps for the User**

1. Access the Flask Application:
   * Open your browser and go to [**http://23.239.18.72**](http://23.239.18.72/)**.**

****

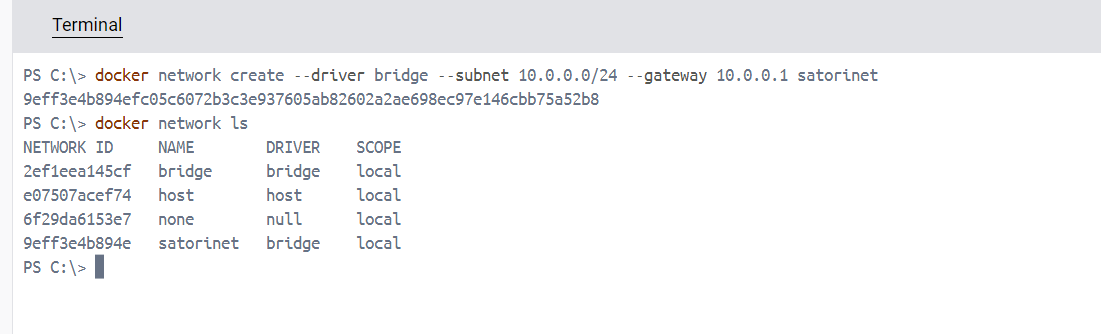
1. Download the Configuration File:
   * After registration, download the configuration file.
2. Install and Import into WireGuard Client:
   * Install the WireGuard client.
   * Import the downloaded configuration file into the client.
3. Start the WireGuard Client:
   * Activate the VPN connection using the WireGuard client.

### **Post-VPN Setup: Creating a Docker Network and Running the Satori Containers**

After successfully connecting to the WireGuard VPN client, follow the steps below to set up the Docker network and run the required containers:

1. Create a Docker Network: Open Command Prompt as Administrator and run the following command to create the Docker network:

docker network create --driver bridge --subnet 10.0.0.0/24 --gateway 10.0.0.1 satori



1. Create Necessary Directories: Navigate to the root directory and create a directory for the Satori project:

cd /

mkdir Satori

cd Satori

1. Clone the Required Repositories: Run the following commands to clone the necessary repositories:

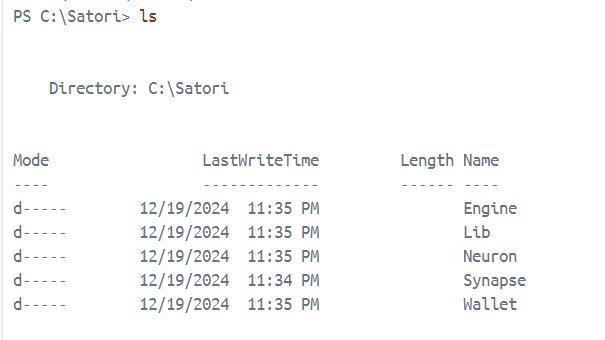
git clone https://github.com/SatoriNetwork/Synapse.git

git clone https://github.com/SatoriNetwork/Lib.git

git clone https://github.com/SatoriNetwork/Wallet.git

git clone https://github.com/SatoriNetwork/Engine.git

git clone https://github.com/SatoriNetwork/Neuron.git



1. Create Additional Directories: Create directories for the Neuron model:

mkdir /Satori/Neuron/models

mkdir /Satori/Neuron/models/huggingface

1. Start the Docker Container: Now, run the following Docker command to start the container, replacing there-ip with the desired IP address for the container:

docker run --rm -it --name satorineuron --net satorinet --ip 10.0.0.6 -p 24601:24601 `

-v C:\Satori\Neuron:/Satori/Neuron `

-v C:\Satori\Synapse:/Satori/Synapse `

-v C:\Satori\Lib:/Satori/Lib `

-v C:\Satori\Wallet:/Satori/Wallet `

-v C:\Satori\Engine:/Satori/Engine `

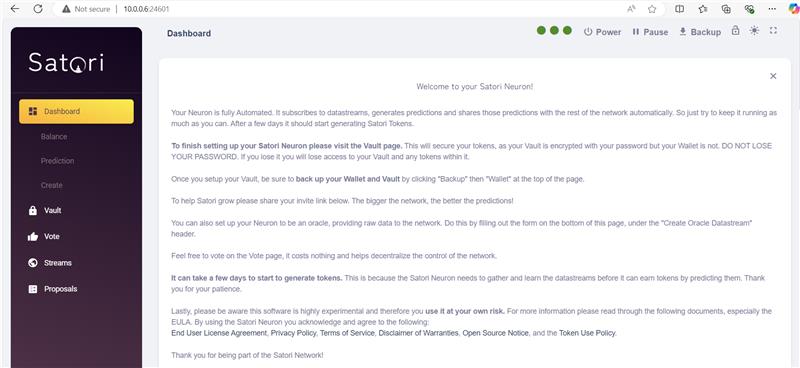
-e ENV=prod `

-e RUNMODE=normal `

satorinet/satorineuron:latest bash

1. Run the Python Script in the Container: Inside the running container, execute the following command to run the Satori Python script:

python satori.py



This will initialize the Satori services within the Docker container. You are now set up and running!