Try V2Ray: A Step-by-Step Guide

V2Ray is a versatile platform for building proxies to bypass network restrictions and enhance online privacy. In this guide, we'll walk you through installing and configuring V2Ray on an Ubuntu server. We'll cover installation steps, configuration files, common issues, and verification methods to ensure everything runs smoothly.

Table of Contents

Installation

Begin by downloading and installing V2Ray using the provided installation script.

```
ubuntu@ip-172-26-0-236:~$ curl -L https://raw.githubusercontent.com/v2fly/fhs-install-v2ray/master/inst
```

Run the Installation Script:

```
chmod +x in.sh
sudo ./in.sh
```

Installation Output:

```
[Install]
WantedBy=multi-user.target


info: V2Ray v5.22.0 is installed.
```

Note: The script suggests removing dependent software if necessary:

```
apt purge curl unzip
```

Configuration

V2Ray Configuration (`config.json`)

This JSON file defines the inbound and outbound settings for V2Ray.

```
{
    "inbounds": [
        {
            "port": 1080,
            "listen": "0.0.0.0",
            "protocol": "vmess",
            "settings": {
                "clients": [
                    {
                        "id": "9f02f6b2-1d7d-4b10-aada-69e050f1be6b",
                        "level": 0,
                        "alterId": 0,
                        "email": "example@v2ray.com",
                        "security": "auto"
                    }
                ]
            },
            "streamSettings": {
                "network": "tcp"
            },
            "sniffing": {
                "enabled": true,
                "destOverride": [
                    "http",
                    "tls"
                ]
            },
            "tag": "vmess-inbound",
            "udp": true
        }
    ],
    "outbounds": [
```

```
        {
            "protocol": "freedom",
            "settings": {},
            "tag": "outbound-freedom",
            "udp": true
        }
    ],
    "log": {
        "loglevel": "debug",
        "access": "/var/log/v2ray/access.log",
        "error": "/var/log/v2ray/error.log"
    },
    "stats": {
        "enabled": false
    },
    "environment": {
        "v2ray.vmess.aead.forced": "false"
    }
}
```

Key Points: - **Inbounds:** Defines the entry points for incoming connections. Here, it's set to use the `vmess` protocol on port `1080`. - **Outbounds:** Specifies where the traffic should be sent. The `freedom` protocol allows traffic to pass without restrictions. - **Logging:** Configured to log access and error information for debugging purposes. - **Security:** The `security` field is set to `aes-256-gcm` for enhanced encryption.

Proxy Configuration (`config.yaml`)

This YAML file configures the proxy settings, DNS, and rules for traffic routing.

```
port: 7890
socks-port: 7891
mixed-port: 7892
allow-lan: true
mode: Rule
log-level: info
external-controller: 0.0.0.0:9090
experimental:
  ignore-resolve-fail: true
```

```yaml
dns:
  enable: false
  listen: 0.0.0.0:53
  enhanced-mode: fake-ip
  fake-ip-range: 198.18.0.1/16
  default-nameserver:
    - 119.29.29.29
    - 223.5.5.5
  nameserver:
    - https://223.5.5.5/dns-query
    - https://1.12.12.12/dns-query
  fake-ip-filter:
    - "*.lan"
    - "*.localdomain"
    - "*.example"
    - "*.invalid"
    - "*.localhost"
    - "*.test"
    - "*.local"

proxies:
  - name: "My VMess Proxy"
    type: vmess
    server: 54.254.0.0
    port: 1080
    uuid: "9f02f6b2-1d7d-4b10-aada-0000"
    alterId: 0
    cipher: "aes-128-gcm"
    udp: true

proxy-groups:
  - name: "Proxy"
    type: select
    proxies:
      - "My VMess Proxy"
```

```
rules:
  - IP-CIDR,192.168.0.0/16,DIRECT
  - IP-CIDR,10.0.0.0/8,DIRECT
  - IP-CIDR,127.0.0.0/8,DIRECT
  - GEOIP,CN,DIRECT
  - MATCH,Proxy
```

Key Points: - **Ports:** Configures various ports for HTTP, SOCKS, and mixed traffic. - **DNS:** Sets up DNS settings with fake IP ranges and specified nameservers. - **Proxies:** Defines a VMess proxy with encryption using `aes-128-gcm`. - **Proxy Groups:** Allows selection between different proxy options. - **Rules:** Directs traffic based on IP ranges and geographical locations.

Note: Ensure that the `cipher` in the proxy configuration matches the `security` setting in the `config.json`.

Managing V2Ray Service

After installation and configuration, you need to manage the V2Ray service using `systemctl`.

Enabling and Starting V2Ray

Enable V2Ray to Start on Boot:

```
sudo systemctl enable v2ray
```

Start V2Ray Service:

```
sudo systemctl start v2ray
```

Expected Output:

```
Created symlink /etc/systemd/system/multi-user.target.wants/v2ray.service → /etc/systemd/system/v2ray.s
```

Verify Service Status:

```
sudo systemctl status v2ray
```

Sample Output:

```
 v2ray.service - V2Ray Service
   Loaded: loaded (/etc/systemd/system/v2ray.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-04-27 12:55:00 UTC; 1min 30s ago
Main PID: 14425 (v2ray)
    Tasks: 8 (limit: 4915)
```

```
   Memory: 36.7M
   CGroup: /system.slice/v2ray.service
           14425 /usr/local/bin/v2ray run -config /usr/local/etc/v2ray/config.json
```

Common Issues and Troubleshooting

Authentication Failure When Enabling V2Ray

Error Message:

```
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Authenticating as: Ubuntu (ubuntu)
Password:
polkit-agent-helper-1: pam_authenticate failed: Authentication failure
==== AUTHENTICATION FAILED ====
Failed to enable unit: Access denied
```

Solution:

Ensure you' re using `sudo` to execute commands that require administrative privileges.

Correct Command:

```
sudo systemctl enable v2ray
```

Verification

After starting the V2Ray service, verify that it' s running correctly.

Check Running Processes

```
ps aux | grep v2ray
```

Sample Output:

```
nobody     14425  4.4  8.6 5460552 36736 ?       Ssl  12:55   0:00 /usr/local/bin/v2ray run -config /usr
ubuntu     14433  0.0  0.5   7076  2176 pts/1    S+   12:55   0:00 grep --color=auto v2ray
```

Test Connectivity Using Telnet

```
telnet your_server_ip 1080
```

Expected Behavior:

- If the connection is successful, you'll see a response from the V2Ray service.
- To exit Telnet, press `Ctrl + ]` and then type `quit`.

Conclusion

Setting up V2Ray on an Ubuntu server involves installing the software, configuring inbound and outbound settings, managing the service with `systemctl`, and verifying its operation. By following this guide, you should have a functional V2Ray setup that enhances your network privacy and bypasses restrictions effectively.

If you encounter any issues or have questions, feel free to leave a comment below!

Additional Tips

- **Security:** Always ensure your V2Ray UUID and passwords are kept secure.
- **Updates:** Regularly update V2Ray to benefit from the latest features and security patches.
- **Monitoring:** Use logs located at `/var/log/v2ray/` to monitor the performance and troubleshoot issues.

Happy proxying!