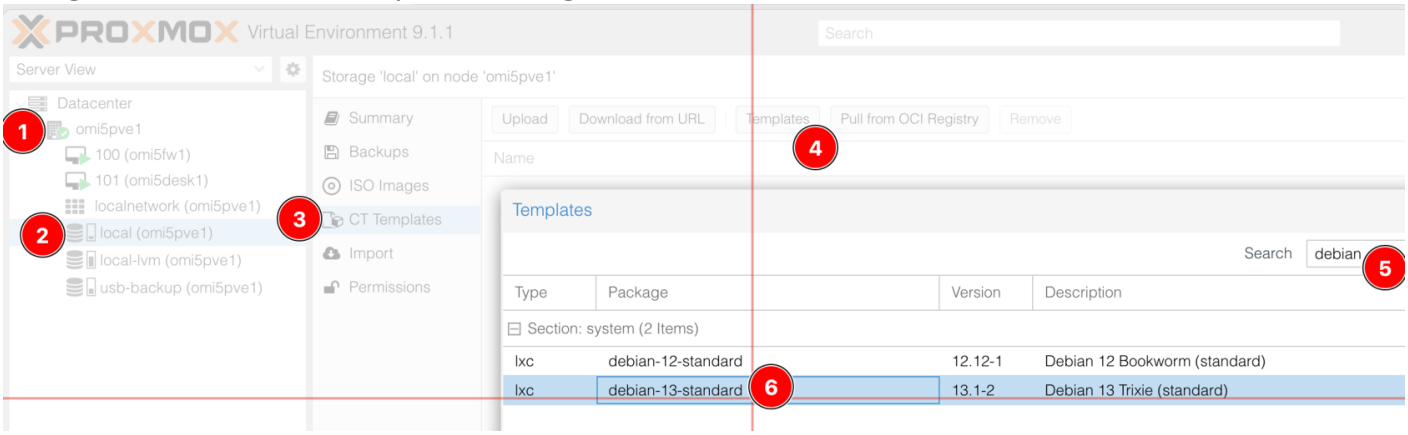
Pi-hole - DNS server

- Pi-hole on container in Proxmox
- Pi-hole - DNS server

Pi-hole on container in Proxmox

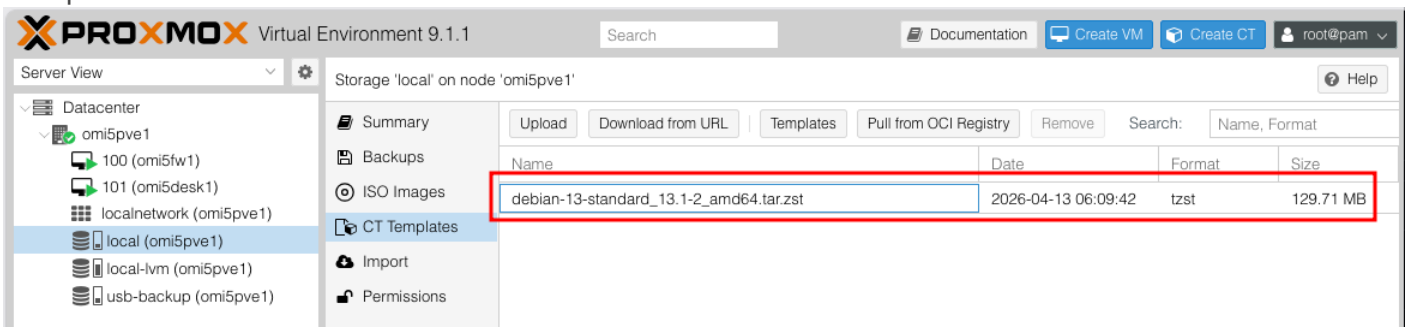
Login into Proxmox

Download Debian distro template for containers. For some reason it might be cancelled or disrupted. As we now, which one template is expecting to find, let's download that manually. Navigate to containers' templates storage, find suitable and hit [Download] button



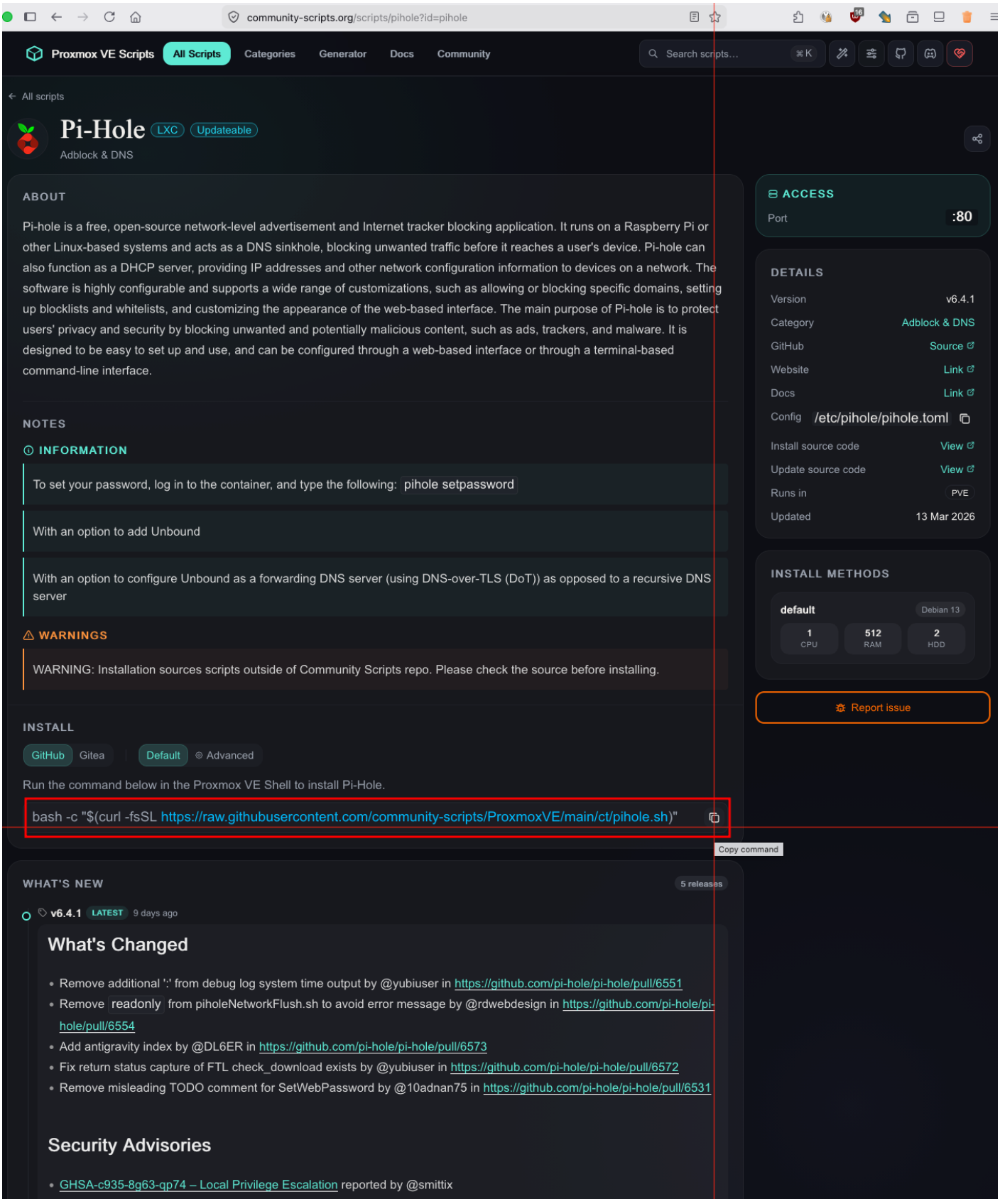
Download task itself

Template downloaded

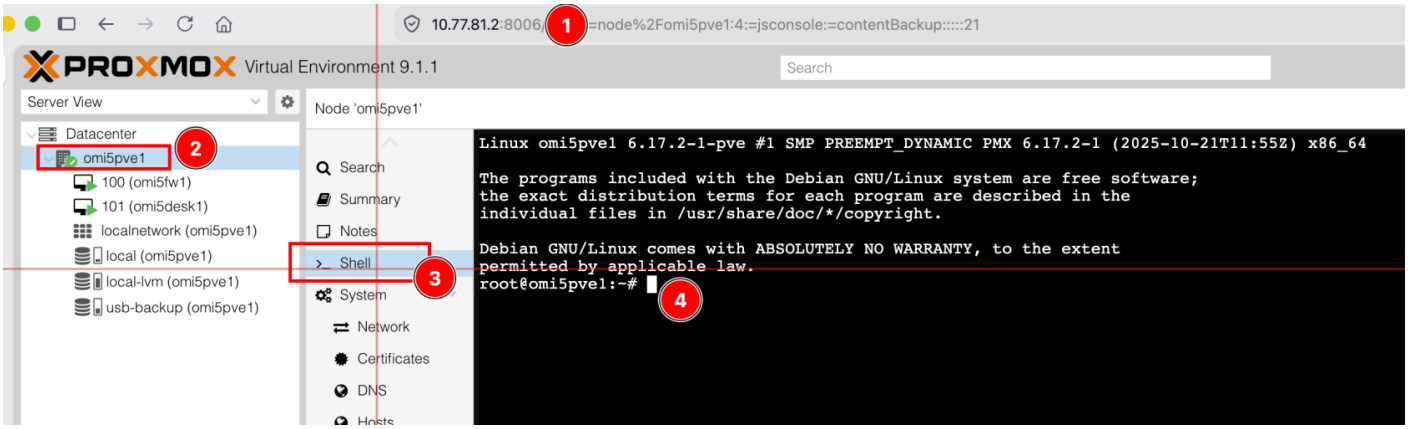


Script location:

```
https://community-scripts.org/scripts/pihole?id=pihole
```

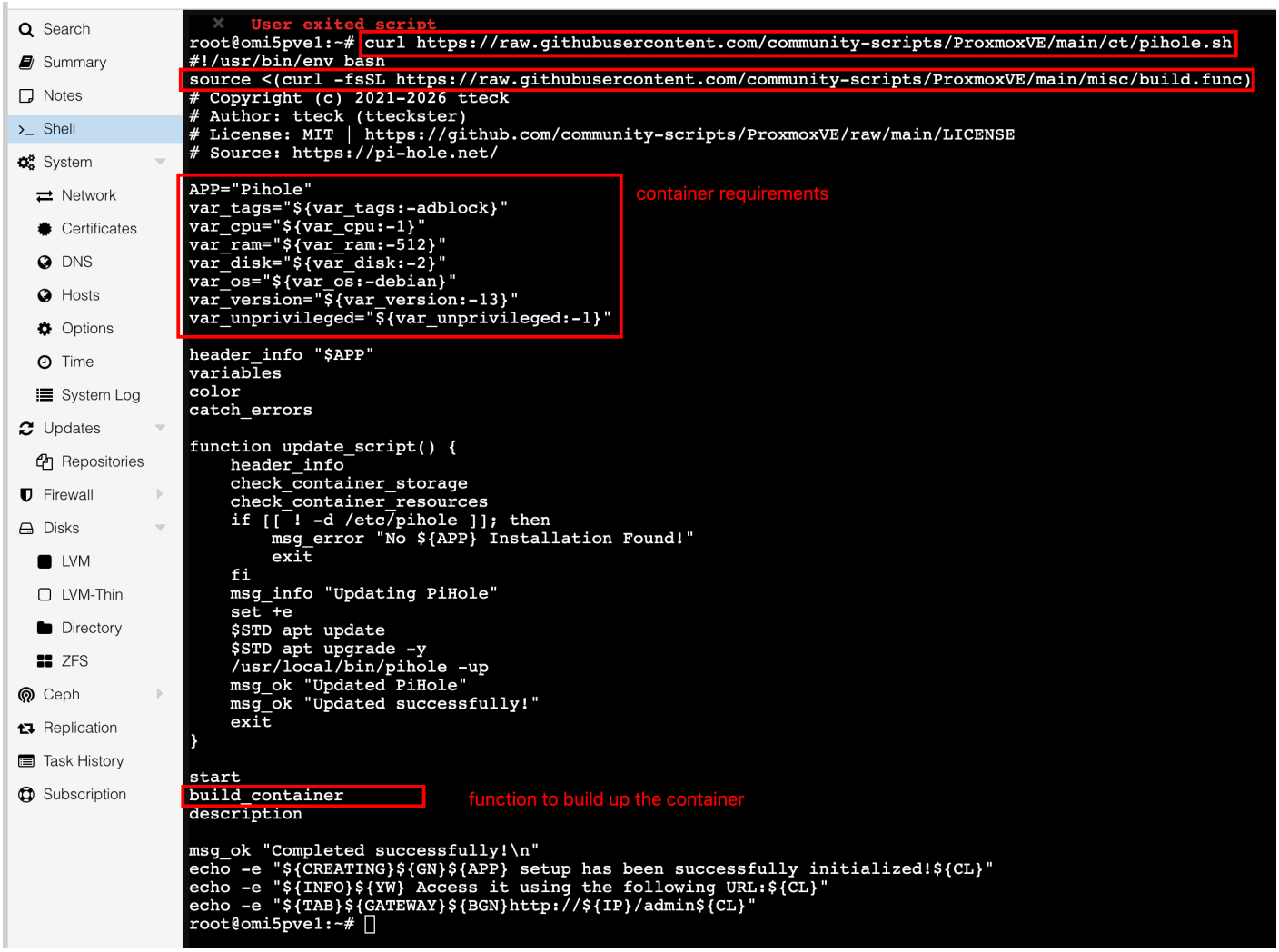


curl https://raw.githubusercontent.com/community-scripts/ProxmoxVE/main/ct/pihole.sh



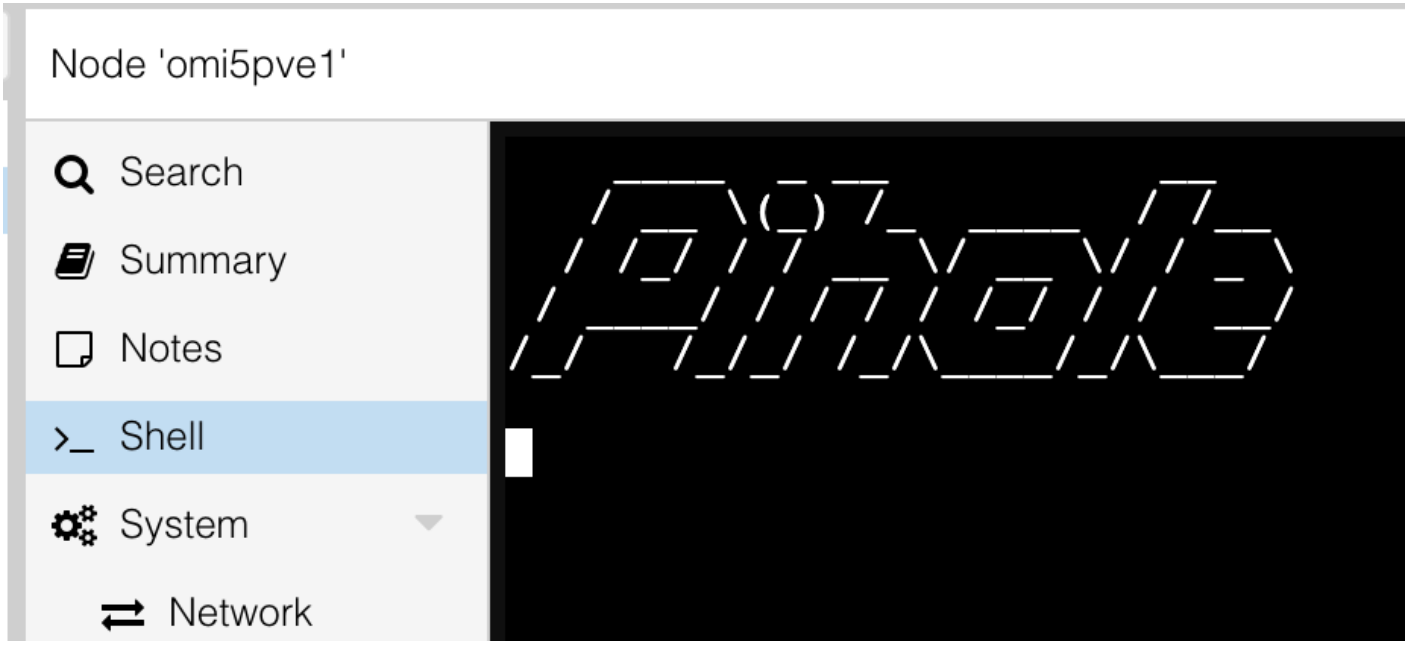
Paste and execute, this will download a script and execute it on hypervisor:

Let's see what is inside of this script

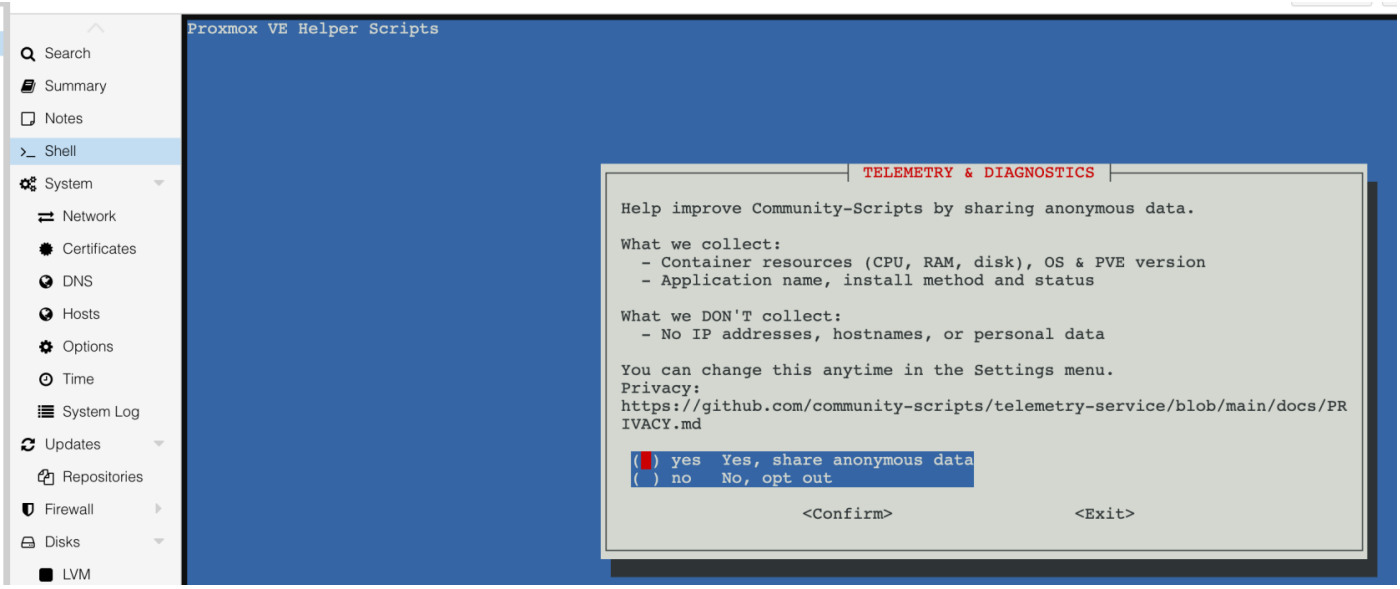


and invoke it's from the shell, when confident:

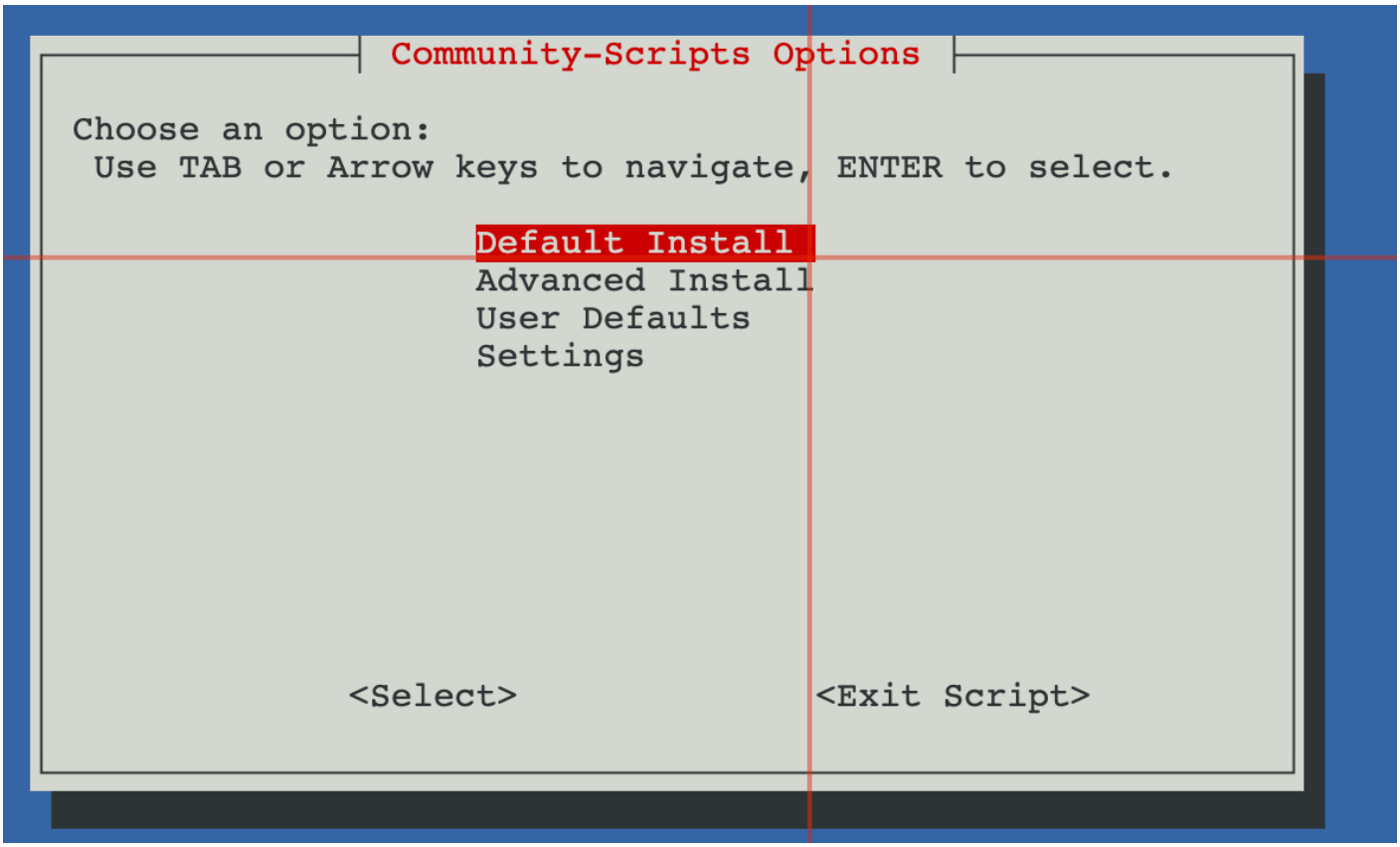
```
bash -c "$(curl -fsSL https://raw.githubusercontent.com/community-scripts/ProxmoxVE/main/ct/pihole.sh)"
```



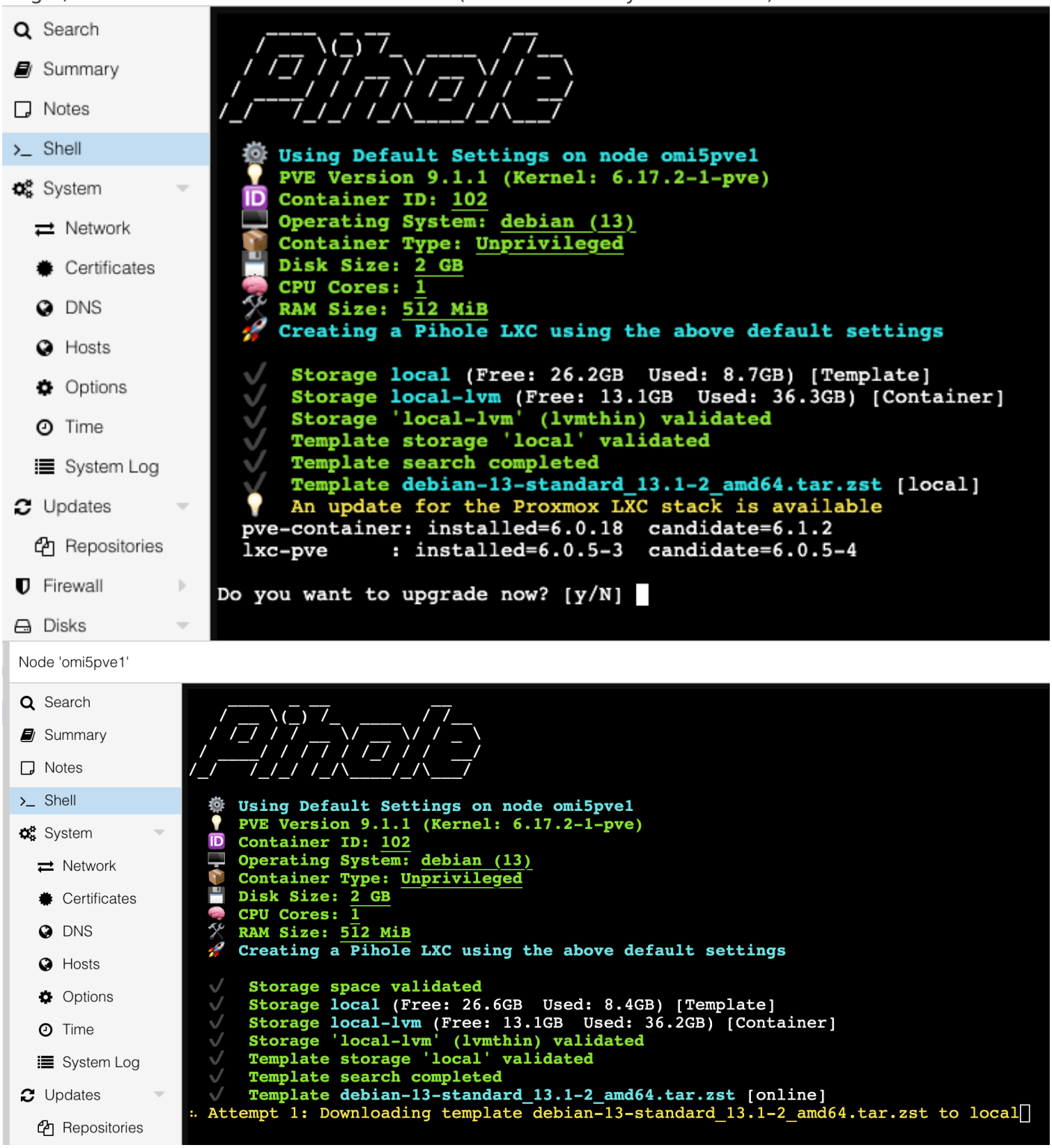
First screen is regarding telemetry data, do you choice using [Tab], [Space], [Enter]:



Use 'Default Install'

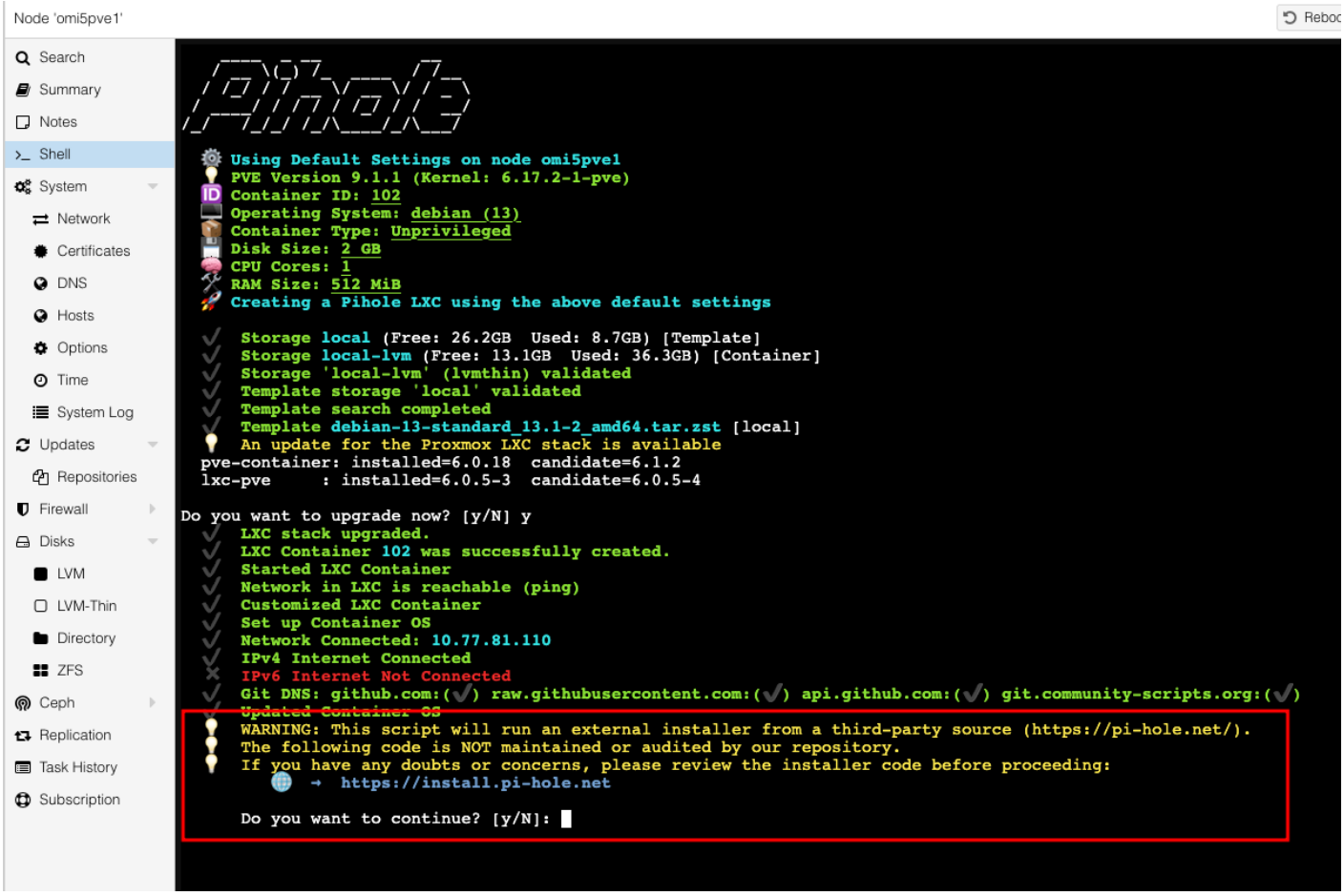


Begin, this will look for Debian Linux distro (and if necessary download it):



Script will download container and do configuration of container

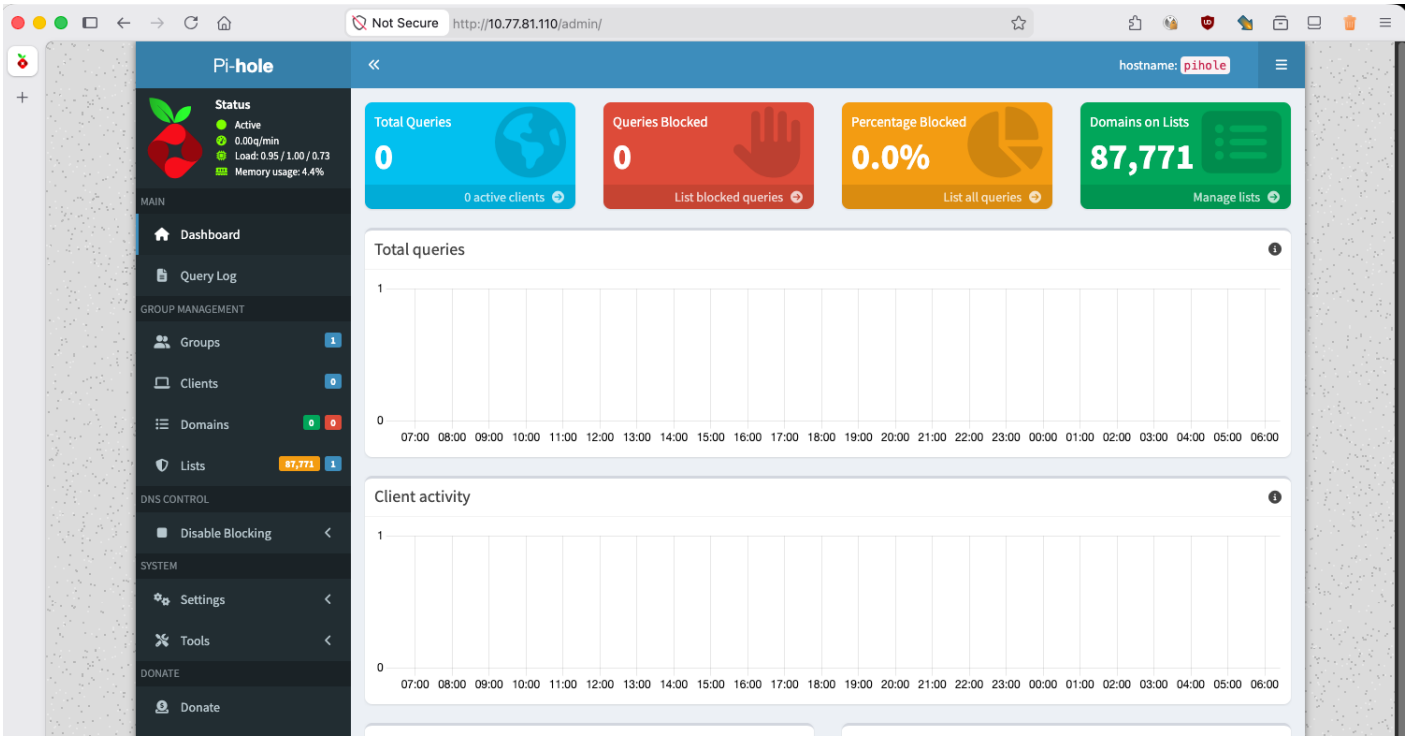
Now, when container is ready, Proxmox's script will invoke Pi-hole script (which is actual application installer)



Final question is about 'Unbound' service

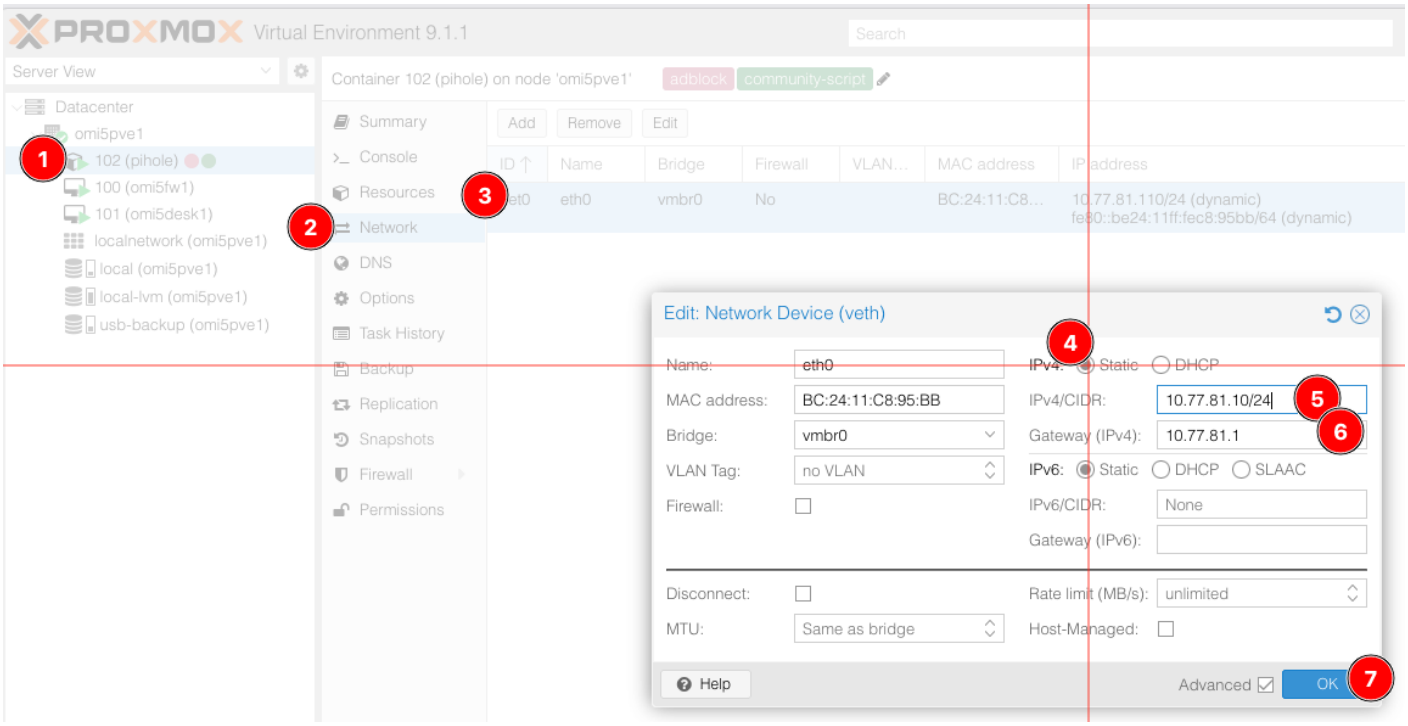
Check in the browser the URL, which is indicated by installer.

```
http://10.77.81.110/admin
```



Configuration of Pi-hole is described in another page.

The only change we want to perform is to change IP address. Locate container, open it's network configuration, change from DHCP to STATIC and define desired IP address (remember do define CIDR mask).



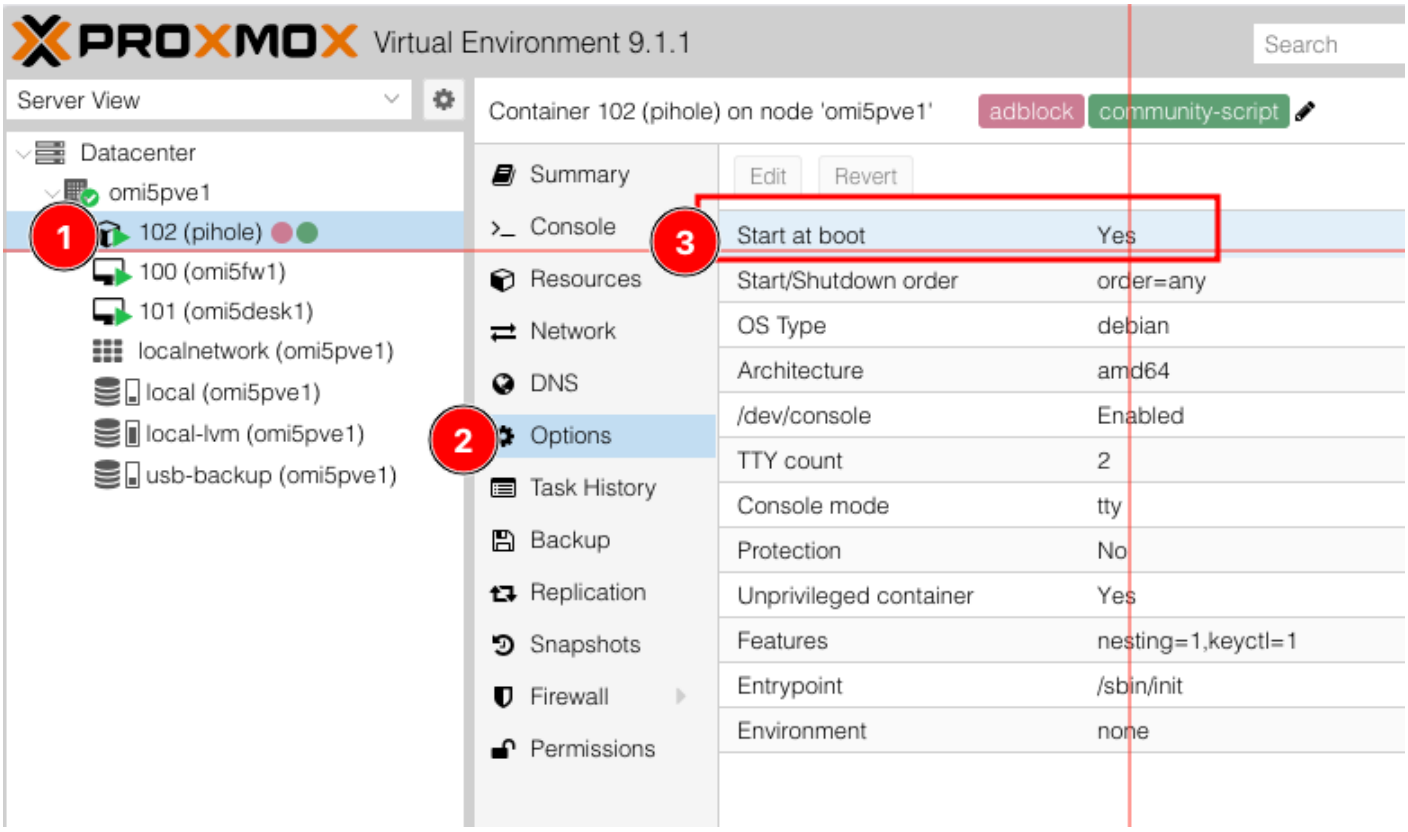
Open container's shell , take a note of network address and perform restart

```
ip -br a
shutdown -r now
```

Verify after reboot again

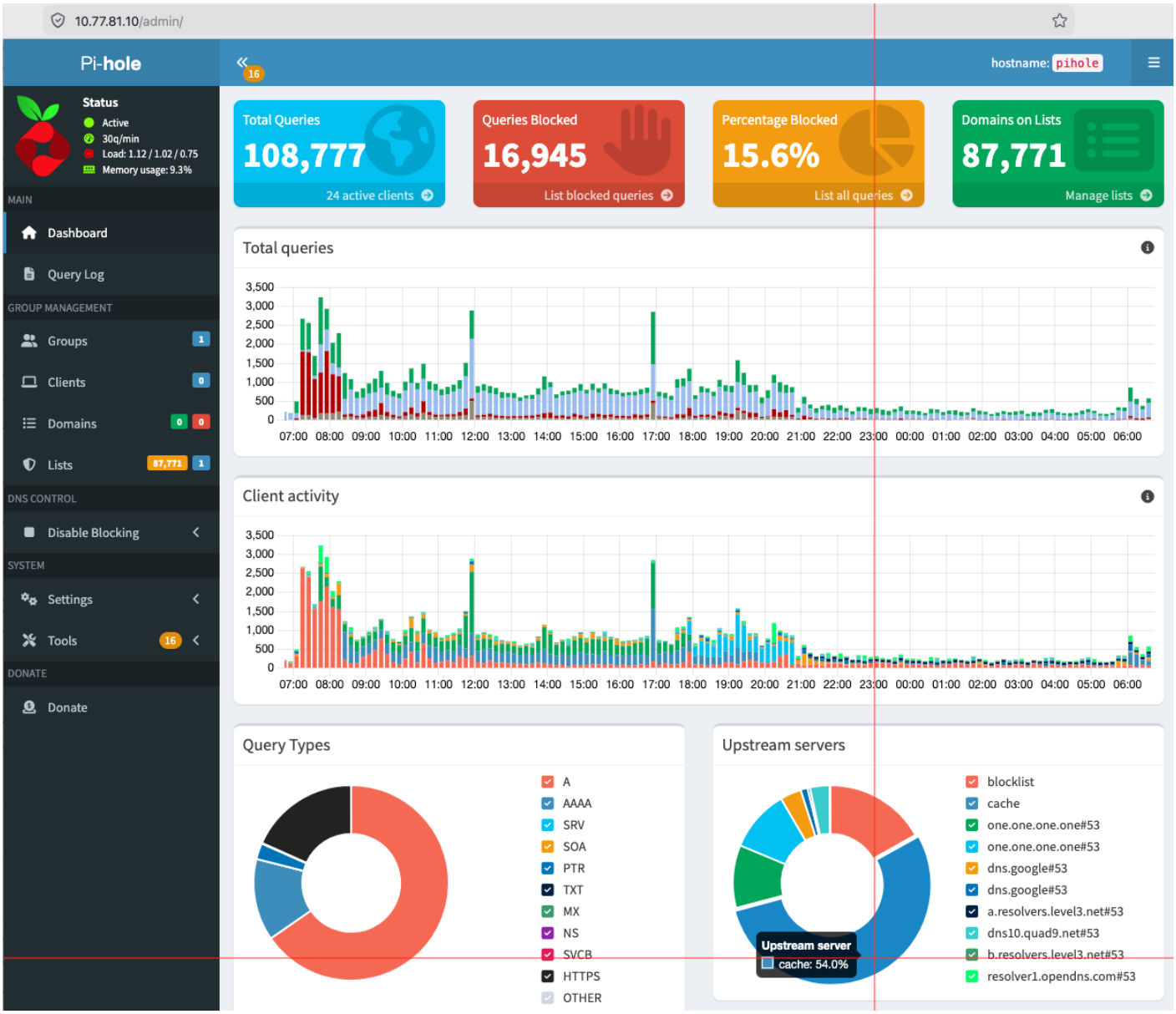
```
Pihole LXC Container
    Provided by: community-scripts ORG | GitHub: https://github.com/community-scripts/ProxmoxVE

    OS: Debian GNU/Linux - Version: 13
    Hostname: pihole
    IP Address: 10.77.81.10
root@pihole:~# ip -br a
lo               UNKNOWN        127.0.0.1/8 ::1/128
eth0@if14        UP             10.77.81.10/24 fe80::be24:11ff:fec8:95bb/64
root@pihole:~#

Pihole LXC Container
    Provided by: community-scripts ORG | GitHub: https://github.com/community-scripts/ProxmoxVE

    OS: Debian GNU/Linux - Version: 13
    Hostname: pihole
    IP Address: 10.77.81.110
root@pihole:~#
root@pihole:~#
root@pihole:~#
root@pihole:~# ip -br a
lo               UNKNOWN        127.0.0.1/8 ::1/128
eth0@if13        UP             10.77.81.110/24 10.77.81.10/24 fe80::be24:11ff:fec8:95bb/64
root@pihole:~# shutdown -r now
```

Last thing, before go: verify that container has "Start on boot" tick box enabled. Thus, we shall be sure that DNS server will be started with hypervisor. That applies to all machines, which need to be started (such as main router :)



Next morning, after the first day of DNS traffic

With no filtering lists loads, DNS server serverd 54% of cached traffic to local clients. That is very good, which means, that local clients received DNS response earlier, rather then for original query.


10.77.81.10/admin/
Pi-hole
16
hostname: pihole
Status
Active
30q/min
Load: 1.12 / 1.02 / 0.75
Memory usage: 9.3%
MAIN
Dashboard
Query Log
GROUP MANAGEMENT
Groups 1
Clients 0
Domains 0 0
Lists 87,771 1
DNS CONTROL
Disable Blocking
SYSTEM
Settings
Tools 16
DONATE
Donate
Total Queries
108,777
24 active clients
Queries Blocked
16,945
List blocked queries
Percentage Blocked
15.6%
List all queries
Domains on Lists
87,771
Manage lists
Total queries
3,500
3,000
2,500
2,000
1,500
1,000
500
0
07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00 23:00 00:00 01:00 02:00 03:00 04:00 05:00 06:00
Client activity
3,500
3,000
2,500
2,000
1,500
1,000
500
0
07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00 23:00 00:00 01:00 02:00 03:00 04:00 05:00 06:00
Query Types
A
AAAA
SRV
SOA
PTR
TXT
MX
NS
SVCB
HTTPS
OTHER
Upstream servers
blocklist
cache
one.one.one.one#53
one.one.one.one#53
dns.google#53
dns.google#53
a.resolvers.level3.net#53
dns10.quad9.net#53
b.resolvers.level3.net#53
resolver1.opendns.com#53
Upstream server
cache: 54.0%

Pi-hole - DNS server

Purpose: Trusted and Filtering Feature Domain Name Resolution Solution: Pi-hole (https://pi-hole.net/) OS: Debian GNU/Linux 12 (bookworm)

preparations:

setup

```
# Pi-hole Automated Installer
This installer will transform your device into a network-wide ad blocker!
[OK]

# Open Source Software
The Pi-hole is free, but powered by your donations:
https://pi-hole.net/donate/
[OK]
```

Decide and assign static IP address

```
# Static IP Needed
The Pi-hole is a SERVER so it needs a STATIC IP ADDRESS to function properly.

IMPORTANT: If you have not already done so, you must ensure that this device has a static IP.

Depending on your operating system, there are many ways to achieve this, through DHCP reservation, or by manually assigning one.
Please continue when the static addressing has been configured.
[Continue]



# Select Upstream DNS Provider. To use your own, select Custom.
(*)  Cloudflare (DNSSEC)



# Blocklists
Pi-hole relies on third party lists in order to block ads.
You can use the suggestion below, and/or add your own after installation.
```

```
Select 'Yes' to include:
StevenBlack's Unified Hosts List
[Yes]



# Enable Logging
Would you like to enable query logging?
[Yes]



# Select a privacy mode for FTL.                    https://docs.pi-hole.net/ftldns/privacylevels/
(*) 0  Show everything
[Continue]
```

```
# Installation Complete!


Configure your devices to use the Pi-hole as their DNS server using:


IPv4: 10.166.0.2
IPv6: Not Configured
If you have not done so already, the above IP should be set to static.
View the web interface at http://pi.hole/admin:80 or
http://10.166.0.2:80/admin


Your Admin Webpage login password is (superpass)
```

Configure admin dashboard listening port

```
vi /etc/pihole/pihole.toml
```

Let's disable IPv6, disable HTTP and swap HTTPS port

```
#  port = "80o,443os,[::]:80o,[::]:443os"
  port = "445os"
```

Restart service and check port has been changed

```
systemctl restart pihole-FTL
ss -ntap | grep hole
```

Advanced usage

Install database cient

```
apt install sqlite
```

Set pattern

```
# export pattern="dox.installanduse.com"
# export pattern="dox.2dz.fi"
export pattern="ntp"
```

List domains from gravity database (blocking list)

```
sqlite3 /etc/pihole/gravity.db \
    "SELECT domain FROM gravity \
    WHERE domain LIKE '%${pattern}%' \
    COLLATE NOCASE ORDER BY domain;"
```

List schema

```
sqlite3 -noheader /etc/pihole/pihole-FTL.db .schema
```

List queries from pihole-FTL database (log)

```
sqlite3 -noheader /etc/pihole/pihole-FTL.db \
    "SELECT DISTINCT domain
    FROM queries
    WHERE domain LIKE '%${pattern}%'
    ORDER BY domain;"
```

Detail

```
sqlite3 -noheader /etc/pihole/pihole-FTL.db \
    "SELECT DISTINCT d.domain
    FROM query_storage q
    JOIN domain_by_id d ON d.id = q.domain
    WHERE d.domain LIKE '%${pattern}%'
    ORDER BY d.domain;"
```

Query and client

```
sqlite3 -noheader /etc/pihole/pihole-FTL.db \
    "SELECT c.ip || ' ' || d.domain
    FROM query_storage q
    JOIN domain_by_id d ON d.id = q.domain
    JOIN client_by_id c ON c.id = q.client
    WHERE d.domain LIKE '%${pattern}%'
    ORDER BY c.ip, d.domain;"

# can be used by adding suffix " | uniq"
```

Queries by specific client

```
export pattern="dox.2dz.fi"
export src="192.168.0.0"

sqlite3 -noheader /etc/pihole/pihole-FTL.db \
    "SELECT DISTINCT d.domain
    FROM query_storage q
    JOIN domain_by_id d ON d.id = q.domain
    JOIN client_by_id c ON c.id = q.client
    WHERE c.ip = '${src}'
      AND d.domain LIKE '%${pattern}%'
    ORDER BY d.domain;"
```

With count, most frequent

```
sqlite3 -noheader /etc/pihole/pihole-FTL.db \
    "SELECT d.domain, COUNT(*)
    FROM query_storage q
    JOIN domain_by_id d ON d.id = q.domain
    WHERE d.domain LIKE '%${pattern}%'
    GROUP BY d.domain
    ORDER BY COUNT(*) DESC, d.domain;"
```

Only last 24 hours

```
sqlite3 -noheader /etc/pihole/pihole-FTL.db \
    "SELECT DISTINCT d.domain
    FROM query_storage q
    JOIN domain_by_id d ON d.id = q.domain
    WHERE d.domain LIKE '%${pattern}%'
```

```
  AND q.timestamp > unixepoch('now','-1 day')
ORDER BY d.domain;"
```

Detail: timestamp + source IP + domain

```
sqlite3 -noheader /etc/pihole/pihole-FTL.db \
  "SELECT datetime(q.timestamp,'unixepoch','localtime') || ' ' || c.ip || ' ' || d.domain
  FROM query_storage q
  JOIN domain_by_id d ON d.id = q.domain
  JOIN client_by_id c ON c.id = q.client
  WHERE d.domain LIKE '%${pattern}%'
  ORDER BY q.timestamp;"
```