

Domain Name System (DNS)

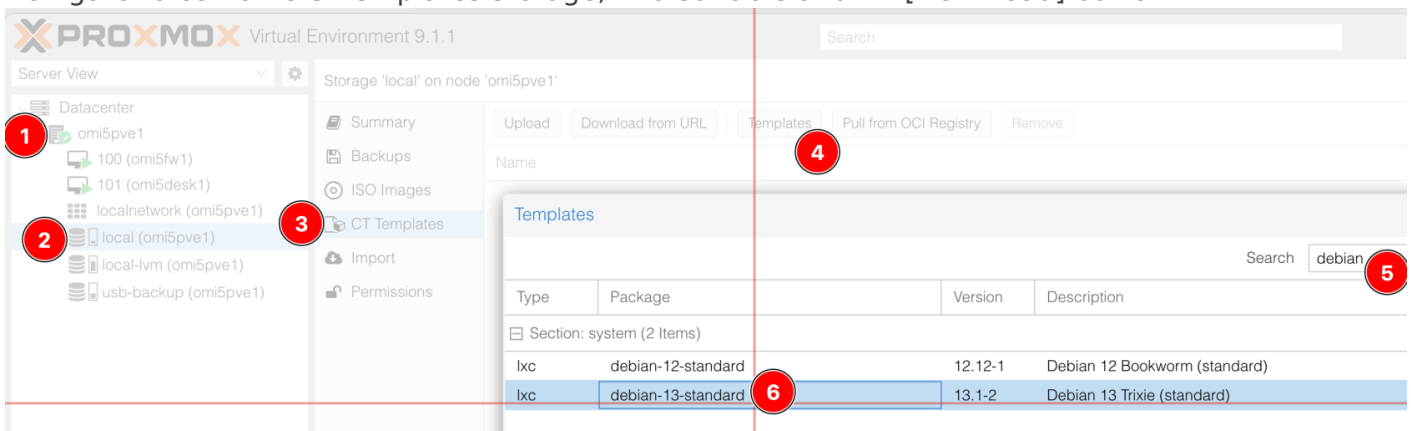
- Pi-hole - DNS server
 - Pi-hole on container in Proxmox
 - Pi-hole - DNS server

Pi-hole - DNS server

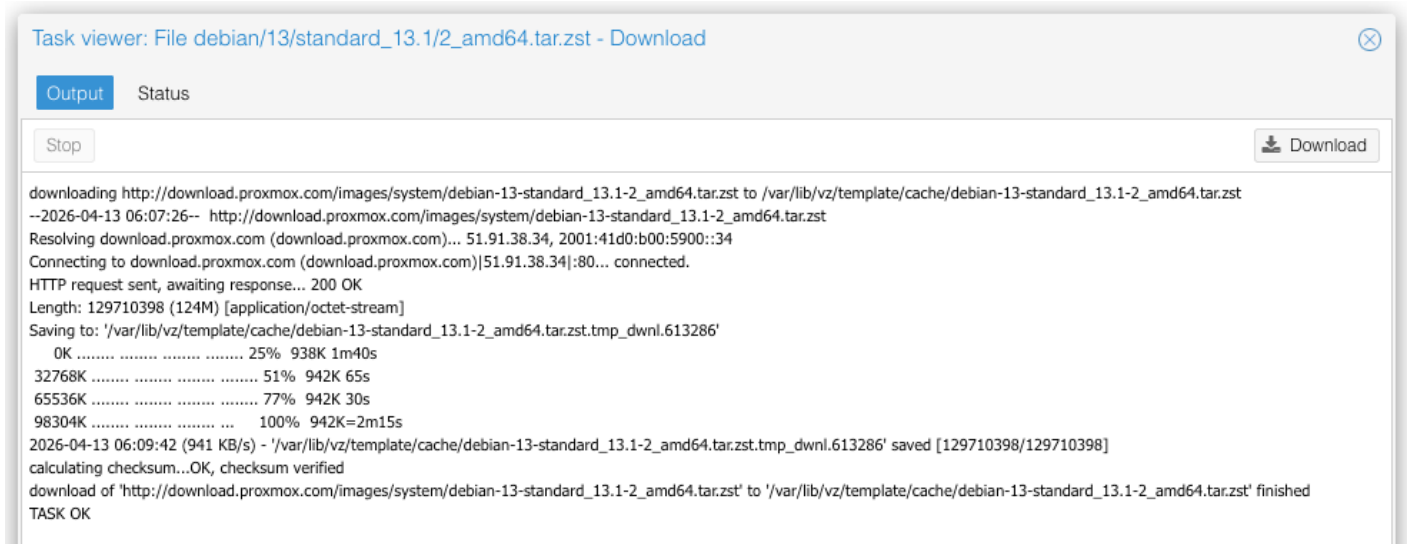
Pi-hole on container in Proxmox

Login into Proxmox

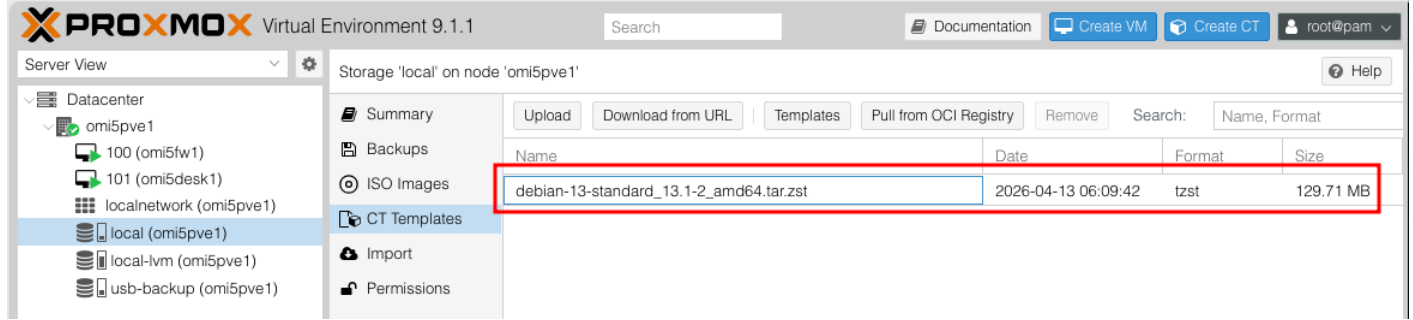
Download Debian distro template for containers. For some reason it might be cancelled or disrupted. As we now, which one template is expecting to find, let's download that manually. Navigate to containers' templates storage, find suitable and hit [Download] button



Download task itself



Template downloaded



The screenshot shows the Proxmox VE 9.1.1 interface. The top bar includes the Proxmox logo, version information, a search bar, and navigation buttons for Documentation, Create VM, Create CT, and the user profile (root@pam). The main content area is titled 'Storage 'local' on node 'omi5pve1'' and features a sidebar with a tree view of the datacenter structure. The 'CT Templates' section is active, displaying a table of templates. A red box highlights the first entry in the table.

| Name | Date | Format | Size |
|---|---------------------|--------|-----------|
| debian-13-standard_13.1-2_amd64.tar.zst | 2026-04-13 06:09:42 | tzst | 129.71 MB |

Script location:

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

community-scripts.org/scripts/pihole?id=pihole

Proxmox VE Scripts All Scripts Categories Generator Docs Community

Search scripts... K

All scripts

Pi-Hole

LXC Updateable

Adblock & DNS

ABOUT

Pi-hole is a free, open-source network-level advertisement and Internet tracker blocking application. It runs on a Raspberry Pi or other Linux-based systems and acts as a DNS sinkhole, blocking unwanted traffic before it reaches a user's device. Pi-hole can also function as a DHCP server, providing IP addresses and other network configuration information to devices on a network. The software is highly configurable and supports a wide range of customizations, such as allowing or blocking specific domains, setting up blocklists and whitelists, and customizing the appearance of the web-based interface. The main purpose of Pi-hole is to protect users' privacy and security by blocking unwanted and potentially malicious content, such as ads, trackers, and malware. It is designed to be easy to set up and use, and can be configured through a web-based interface or through a terminal-based command-line interface.

NOTES

INFORMATION

To set your password, log in to the container, and type the following: `pihole setpassword`

With an option to add Unbound

With an option to configure Unbound as a forwarding DNS server (using DNS-over-TLS (DoT)) as opposed to a recursive DNS server

WARNINGS

WARNING: Installation sources scripts outside of Community Scripts repo. Please check the source before installing.

INSTALL

GitHub Gitea Default Advanced

Run the command below in the Proxmox VE Shell to install Pi-Hole.

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

Copy command

WHAT'S NEW

5 releases

v6.4.1 LATEST 9 days ago

What's Changed

- Remove additional ':' from debug log system time output by @yubiuser in <https://github.com/pi-hole/pi-hole/pull/6551>
- Remove `readonly` from `piholeNetworkFlush.sh` to avoid error message by @rdwebdesign in <https://github.com/pi-hole/pi-hole/pull/6554>
- Add antigravity index by @DL6ER in <https://github.com/pi-hole/pi-hole/pull/6573>
- Fix return status capture of FTL check_download exists by @yubiuser in <https://github.com/pi-hole/pi-hole/pull/6572>
- Remove misleading TODO comment for SetWebPassword by @10adnan75 in <https://github.com/pi-hole/pi-hole/pull/6531>

Security Advisories

- [GHSA-c935-8g63-qp74 – Local Privilege Escalation](#) reported by @smittix

ACCESS

Port :80

DETAILS

Version v6.4.1

Category Adblock & DNS

GitHub Source

Website Link

Docs Link

Config `/etc/pihole/pihole.toml`

Install source code View

Update source code View

Runs in PVE

Updated 13 Mar 2026

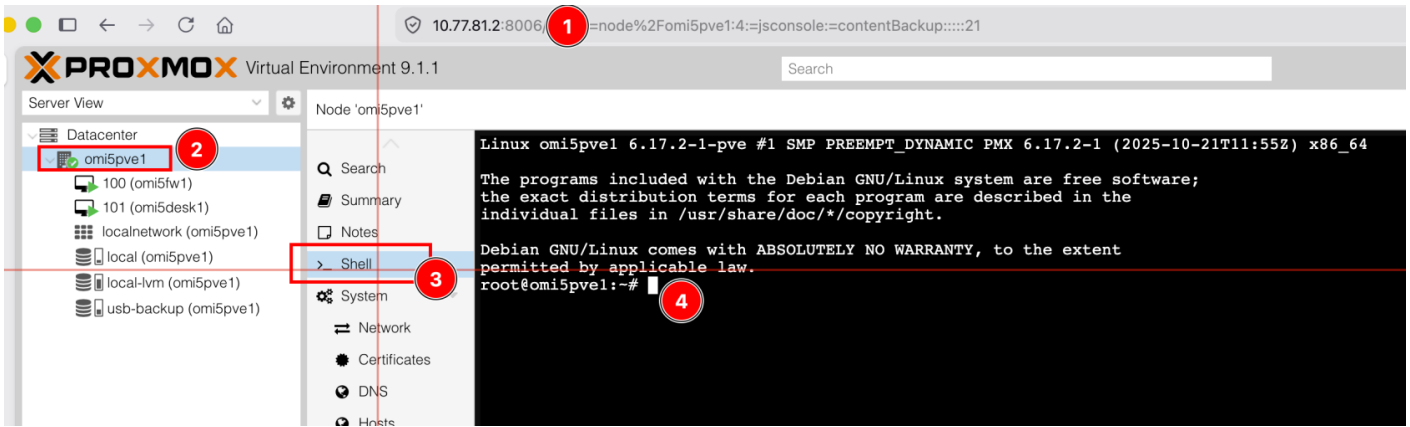
INSTALL METHODS

default Debian 13

1 CPU 512 RAM 2 HDD

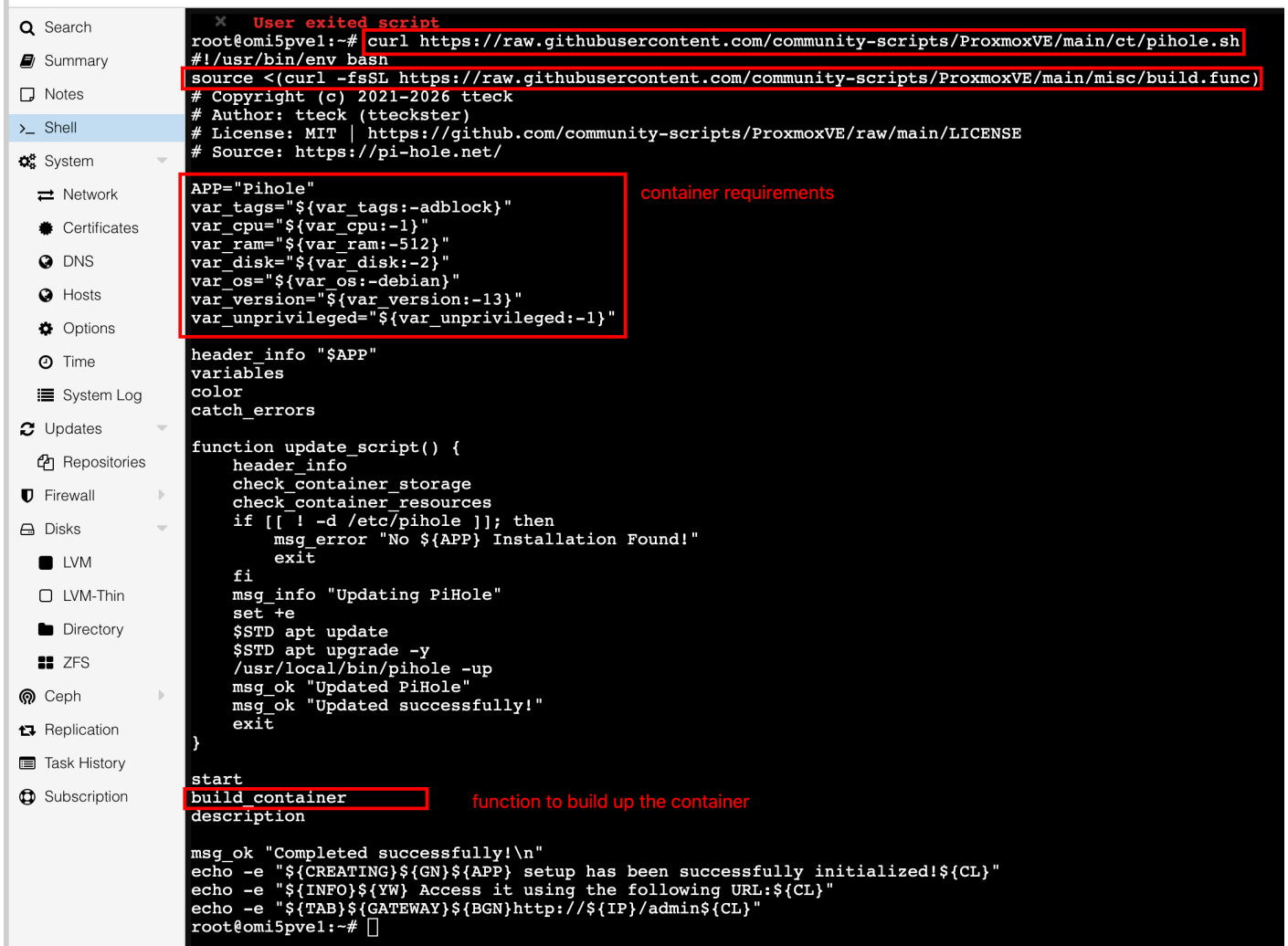
Report issue

```
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)"
```

Node 'omi5pve1'

- Search
- Summary
- Notes
- >_ Shell
- System
- Network



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

Proxmox VE Helper Scripts

TELEMETRY & DIAGNOSTICS

Help improve Community-Scripts by sharing anonymous data.

What we collect:

- Container resources (CPU, RAM, disk), OS & PVE version
- Application name, install method and status

What we DON'T collect:

- No IP addresses, hostnames, or personal data

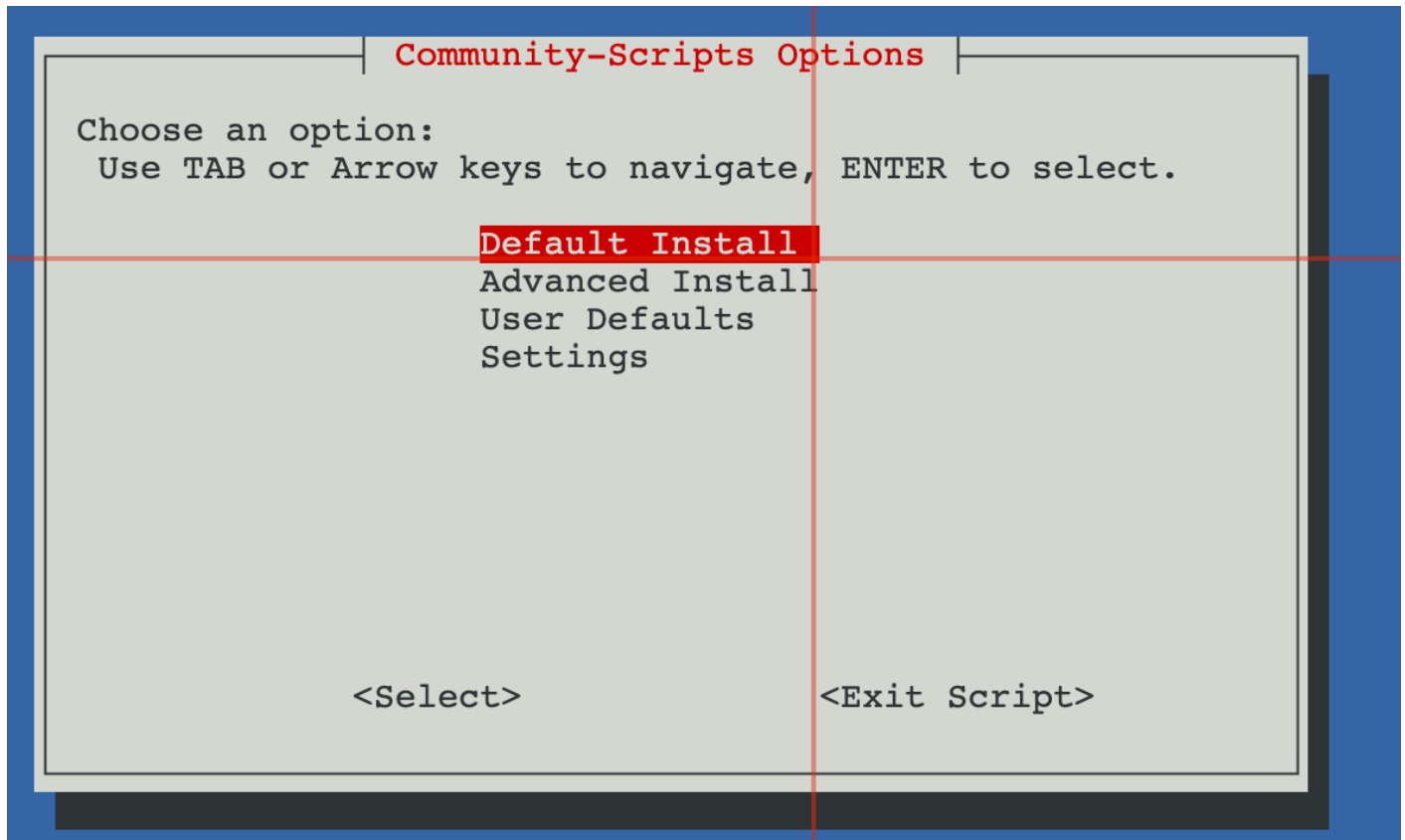
You can change this anytime in the Settings menu.

Privacy:
<https://github.com/community-scripts/telemetry-service/blob/main/docs/PRIVACY.md>

yes Yes, share anonymous data
 no No, opt out

<Confirm> <Exit>

Use 'Default Install'



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

```
Search
Summary
Notes
>_ Shell
System
  Network
  Certificates
  DNS
  Hosts
  Options
  Time
  System Log
Updates
  Repositories
Firewall
Disks
```

```

  Pi-hole

Using Default Settings on node omi5pve1
PVE Version 9.1.1 (Kernel: 6.17.2-1-pve)
Container ID: 102
Operating System: debian (13)
Container Type: Unprivileged
Disk Size: 2 GB
CPU Cores: 1
RAM Size: 512 MiB
Creating a Pi-hole LXC using the above default settings

Storage local (Free: 26.2GB Used: 8.7GB) [Template]
Storage local-lvm (Free: 13.1GB Used: 36.3GB) [Container]
Storage 'local-lvm' (lvmthin) validated
Template storage 'local' validated
Template search completed
Template debian-13-standard_13.1-2_amd64.tar.zst [local]
An update for the Proxmox LXC stack is available
pve-container: installed=6.0.18 candidate=6.1.2
lxc-pve       : installed=6.0.5-3 candidate=6.0.5-4

Do you want to upgrade now? [y/N]
```

```
Node 'omi5pve1'
```

```
Search
Summary
Notes
>_ Shell
System
  Network
  Certificates
  DNS
  Hosts
  Options
  Time
  System Log
Updates
  Repositories
```

```

  Pi-hole

Using Default Settings on node omi5pve1
PVE Version 9.1.1 (Kernel: 6.17.2-1-pve)
Container ID: 102
Operating System: debian (13)
Container Type: Unprivileged
Disk Size: 2 GB
CPU Cores: 1
RAM Size: 512 MiB
Creating a Pi-hole LXC using the above default settings

Storage space validated
Storage local (Free: 26.6GB Used: 8.4GB) [Template]
Storage local-lvm (Free: 13.1GB Used: 36.2GB) [Container]
Storage 'local-lvm' (lvmthin) validated
Template storage 'local' validated
Template search completed
Template debian-13-standard_13.1-2_amd64.tar.zst [online]
.: Attempt 1: Downloading template debian-13-standard_13.1-2_amd64.tar.zst to local
```

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)

```
Node 'omi5pve1' Reboot
Search
Summary
Notes
Shell
System
Network
Certificates
DNS
Hosts
Options
Time
System Log
Updates
Repositories
Firewall
Disks
LVM
LVM-Thin
Directory
ZFS
Ceph
Replication
Task History
Subscription

Using Default Settings on node omi5pve1
PVE Version 9.1.1 (Kernel: 6.17.2-1-pve)
Container ID: 102
Operating System: debian (13)
Container Type: Unprivileged
Disk Size: 2 GB
CPU Cores: 1
RAM Size: 512 MiB
Creating a Pihole LXC using the above default settings

Storage local (Free: 26.2GB Used: 8.7GB) [Template]
Storage local-lvm (Free: 13.1GB Used: 36.3GB) [Container]
Storage 'local-lvm' (lvmthin) validated
Template storage 'local' validated
Template search completed
Template debian-13-standard_13.1-2_amd64.tar.zst [local]
An update for the Proxmox LXC stack is available
pve-container: installed=6.0.18 candidate=6.1.2
lxc-pve : installed=6.0.5-3 candidate=6.0.5-4

Do you want to upgrade now? [y/N] y
LXC stack upgraded.
LXC Container 102 was successfully created.
Started LXC Container
Network in LXC is reachable (ping)
Customized LXC Container
Set up Container OS
Network Connected: 10.77.81.110
IPv4 Internet Connected
IPv6 Internet Not Connected
Git DNS: github.com:(✓) raw.githubusercontent.com:(✓) api.github.com:(✓) git.community-scripts.org:(✓)
Updated Container OS
WARNING: This script will run an external installer from a third-party source (https://pi-hole.net/).
The following code is NOT maintained or audited by our repository.
If you have any doubts or concerns, please review the installer code before proceeding:
  → https://install.pi-hole.net

Do you want to continue? [y/N]:
```

Final question is about 'Unbound' service

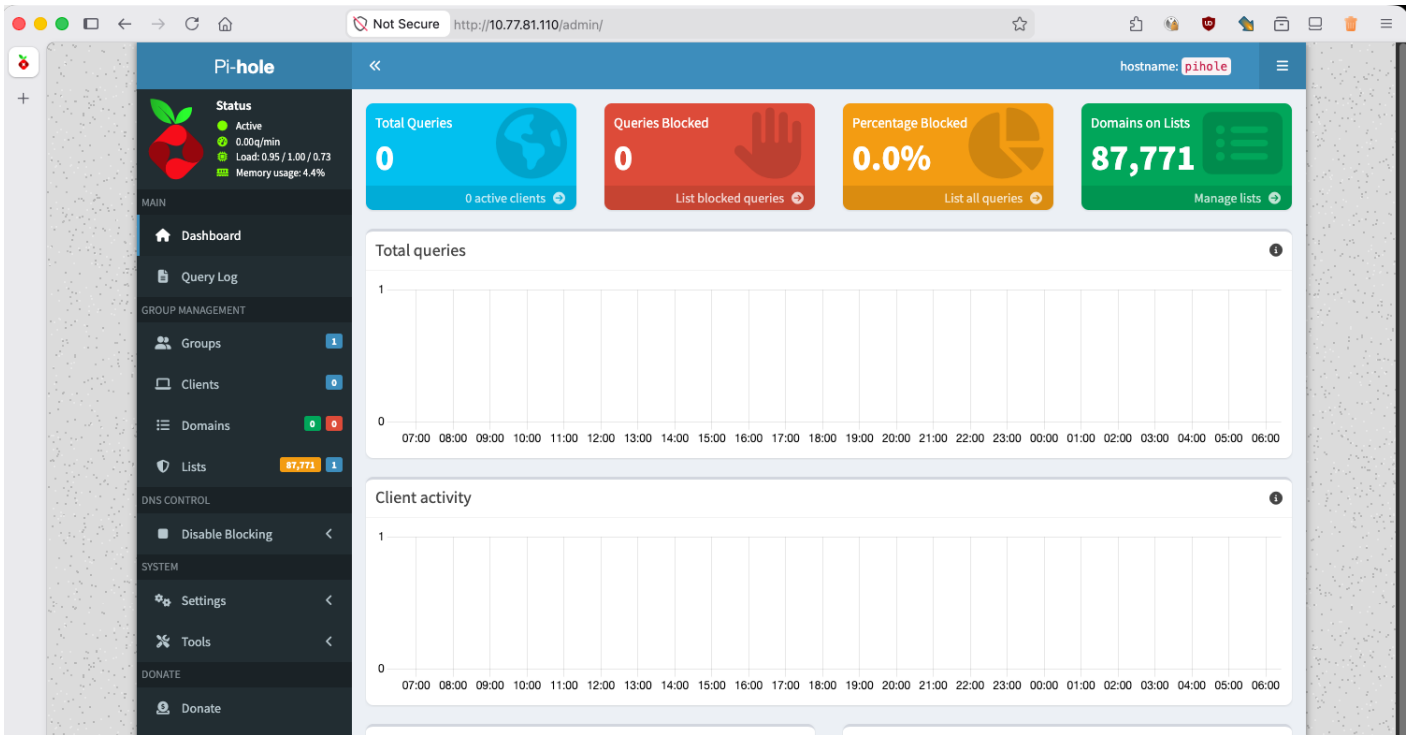
```
Updated Container OS
WARNING: This script will run an external installer from a third-party source (https://pi-hole.net/).
The following code is NOT maintained or audited by our repository.
If you have any doubts or concerns, please review the installer code before proceeding:
  → https://install.pi-hole.net

Do you want to continue? [y/N]: y
Installed Dependencies
Installed Pi-hole
Would you like to add Unbound? <y/N> n
Customized Container
Cleaned
Completed successfully!

Pihole setup has been successfully initialized!
Access it using the following URL:
  http://10.77.81.110/admin
root@omi5pve1:~#
```

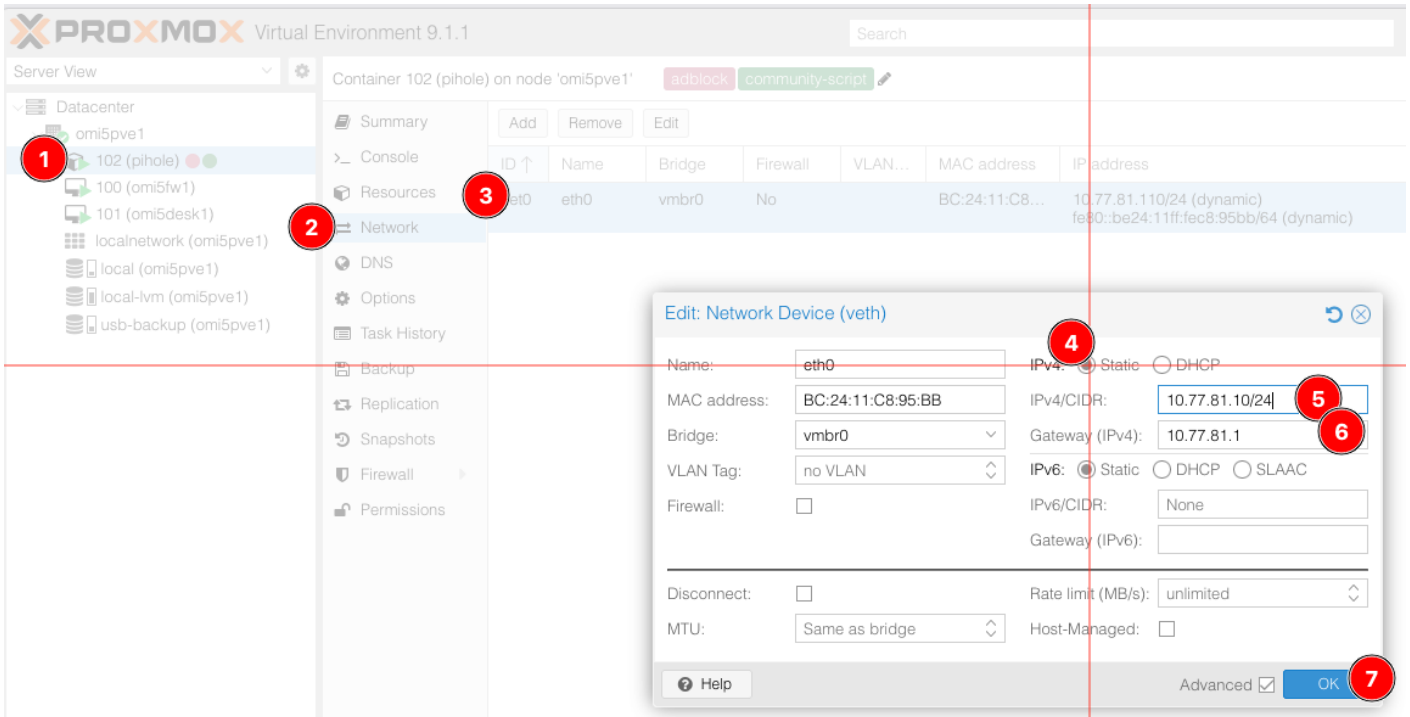
Check in the browser the URL, which is indicated by installer.

<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:~#

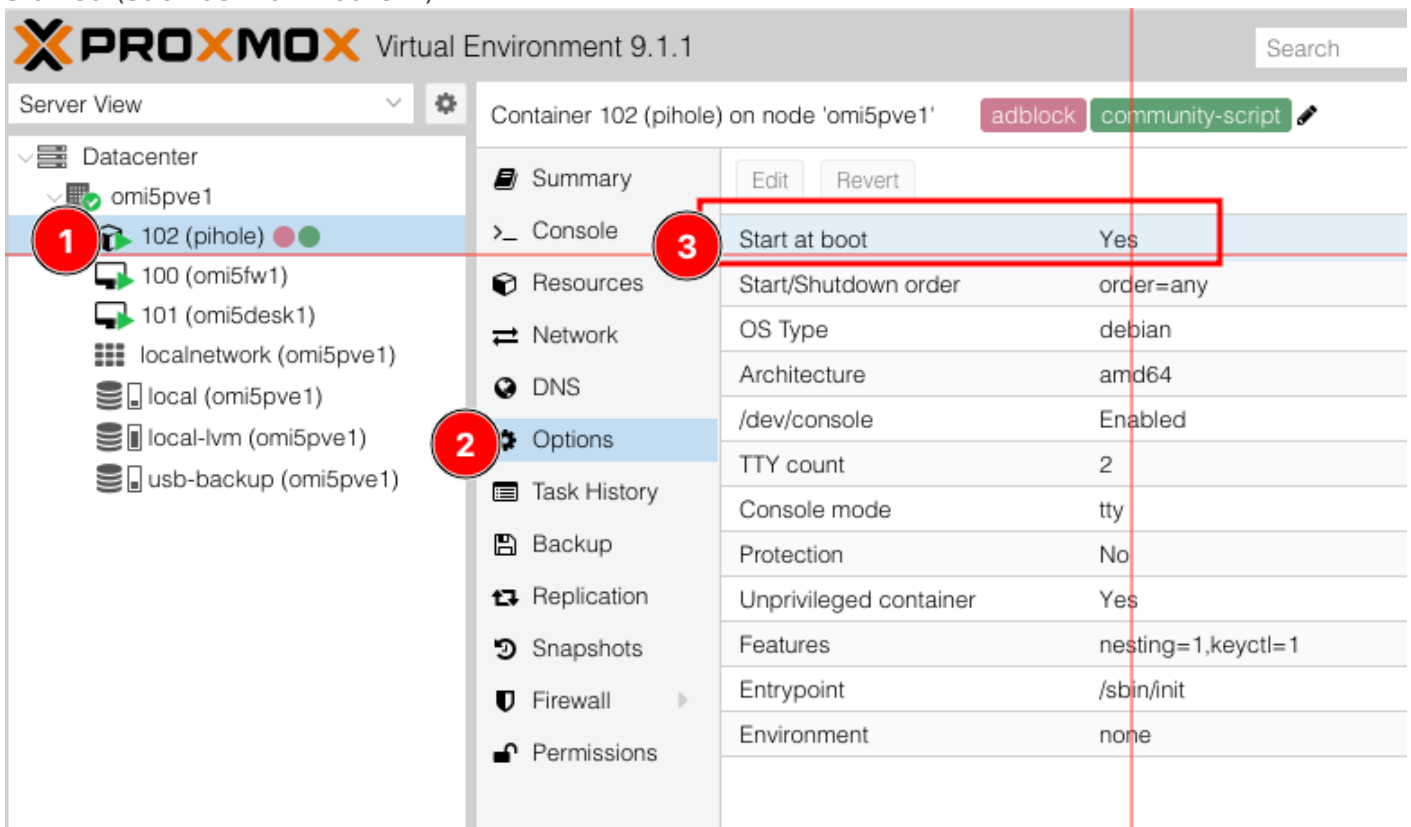
Pihole LXC Container
Provided by: community-scripts ORG | GitHub: https://github.com/community-scripts/ProxmoxVE

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 served 54% of cached traffic to local clients. That is very good, which means, that local clients received DNS response earlier, rather than for original query.



Status

- Active
- 30q/min
- Load: 1.12 / 1.02 / 0.75
- Memory usage: 9.3%

Total Queries

108,777

24 active clients

Queries Blocked

16,945

List blocked queries

Percentage Blocked

15.6%

List all queries

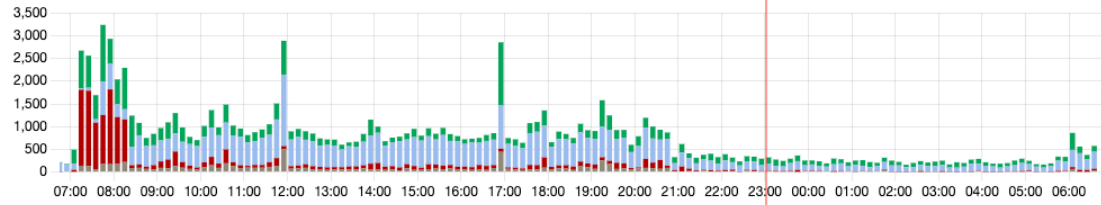
Domains on Lists

87,771

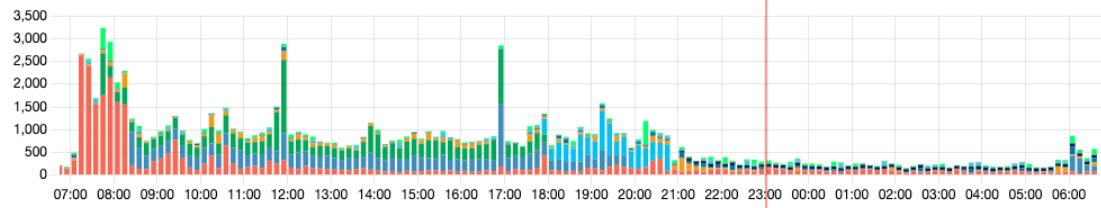
Manage lists

- 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

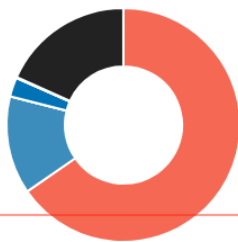
Total queries



Client activity

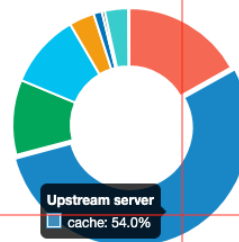


Query Types



- 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
- a.resolvers.level3.net#53
- dns10.quad9.net#53
- b.resolvers.level3.net#53
- resolver1.opendns.com#53

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 client

```
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;"
```