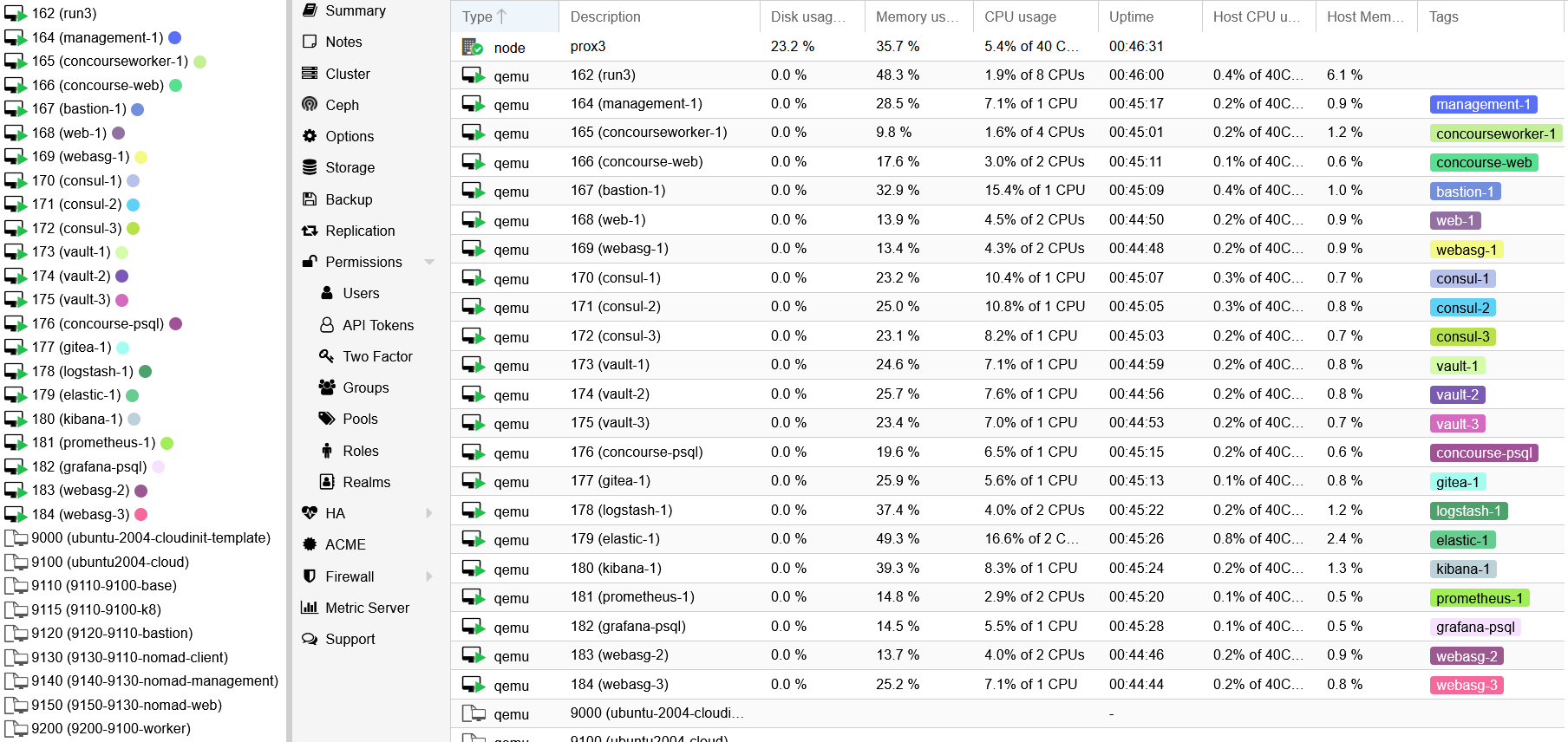
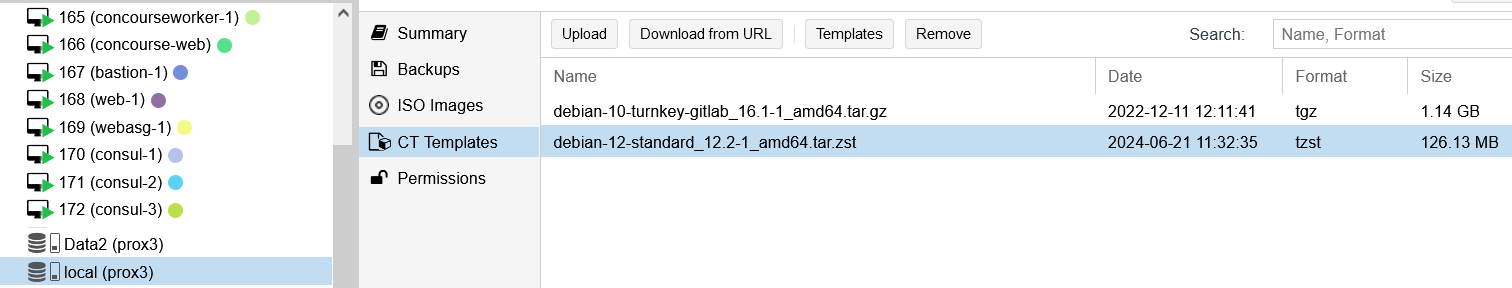
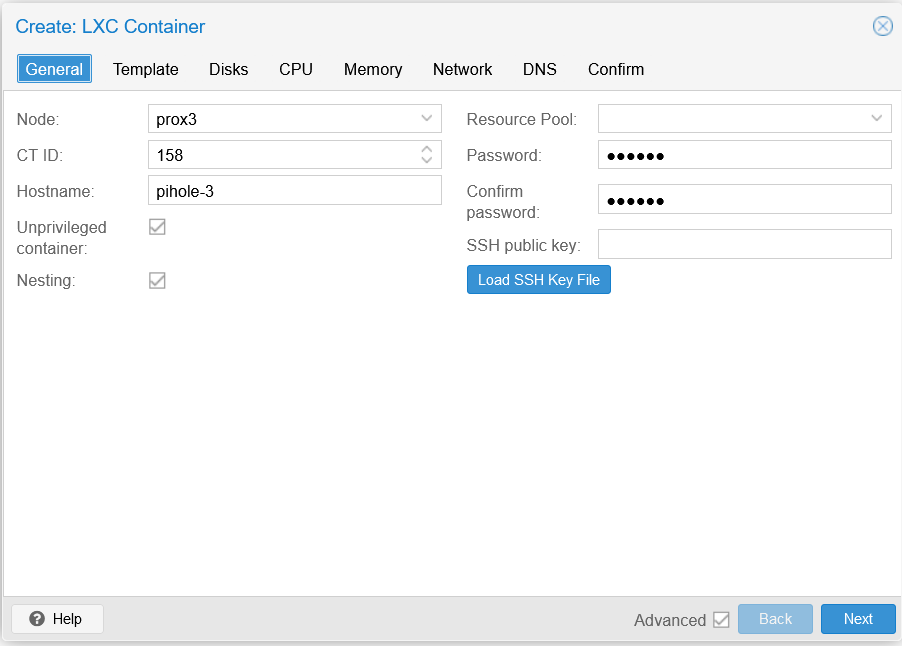
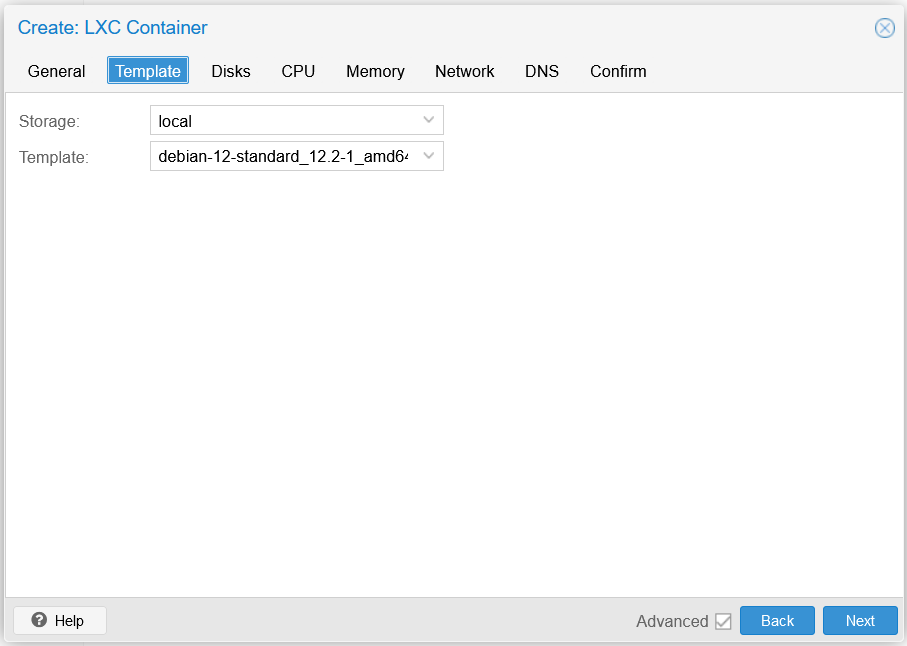
**30 - R710 Proxmox - PiHole for homelab**

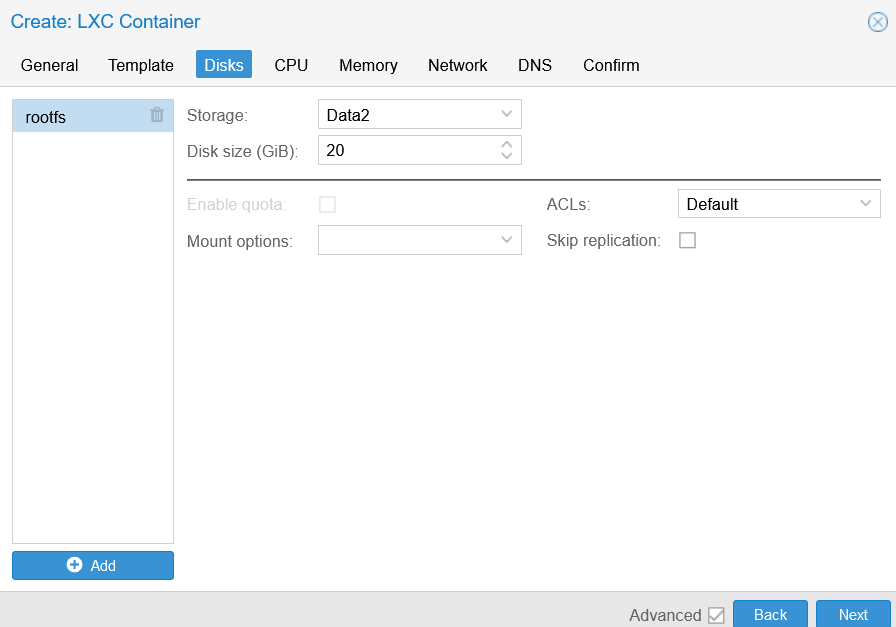
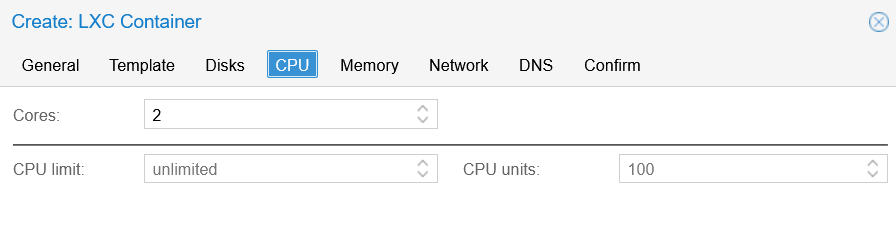
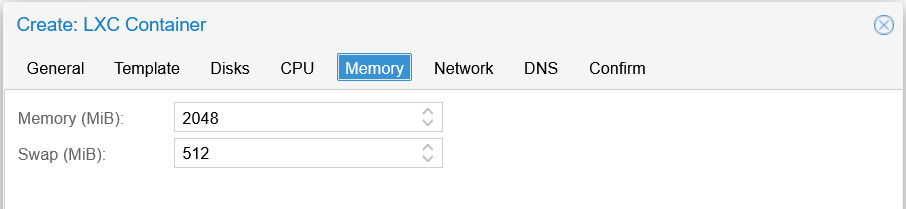
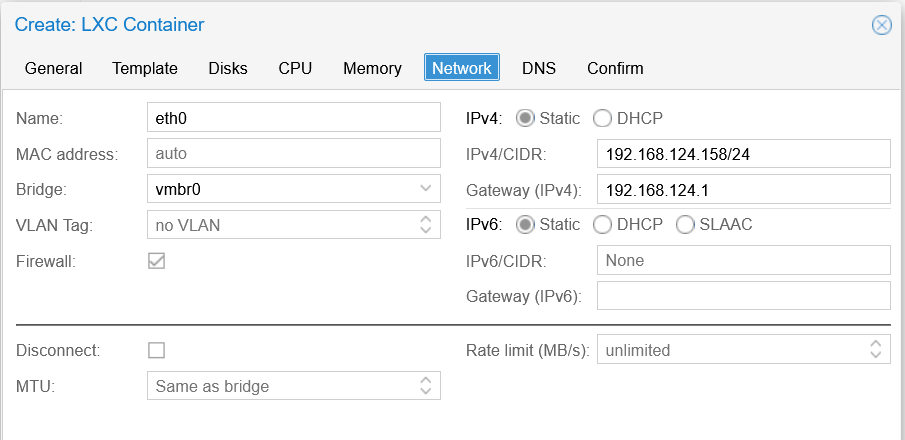
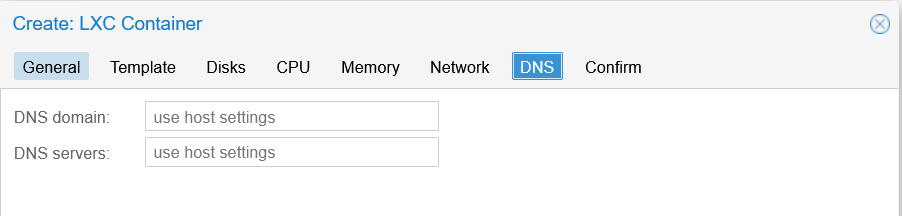
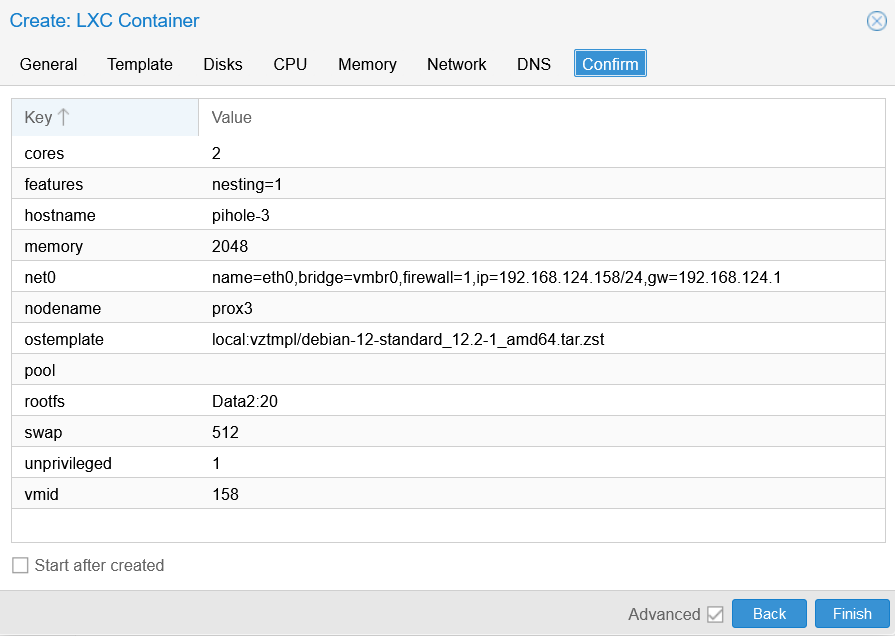
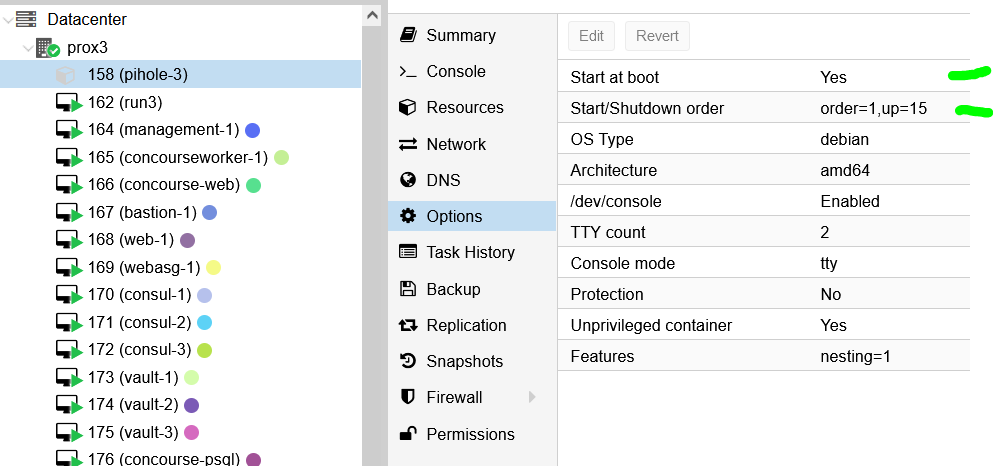
Setup PiHole for name resolution of vm’s in homelab:

****

# 1. Configuring a Debian LXC Container for Pi-hole:

1. Configuring a Debian LXC Container:  
   Select storage location to store the container template, select CT Templates, then select Templates  
   
2. Search for **Debian**, then select **debian-12-standard** and **Download (already downloaded in above picture)**
3. After template has downloaded, select Create CT
4. Enter Hostname, CT ID and Password and click Next  
   
5. Select the Template just downloaded and click Next



1. Adjust Storage placement and Disk Size as below and click Next  
   
2. Adjust CPU details as below and click Next  
     
   \*\* change the above to 1 CPU
3. Adjust Memory details as below and click Next  
     
   \*\* change the above to 1024 Memory
4. Adjust Network details as below and click Next  
   
5. Adjust DNS details as below and click Next  
   
6. Confirm the settings as below and click Finish  
   
7. Adjust Start at boot and order as below:  
   
8. Start the container and once it is running, in the Proxmox Console log in as root with previously set password .. and update the system with:  
     
   **apt update && apt upgrade -y**
9. After the system has updated, install curl with:

**apt install curl -y**

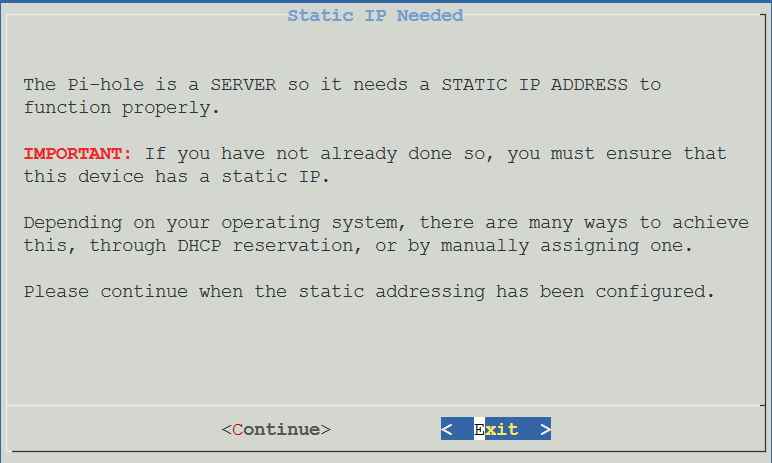
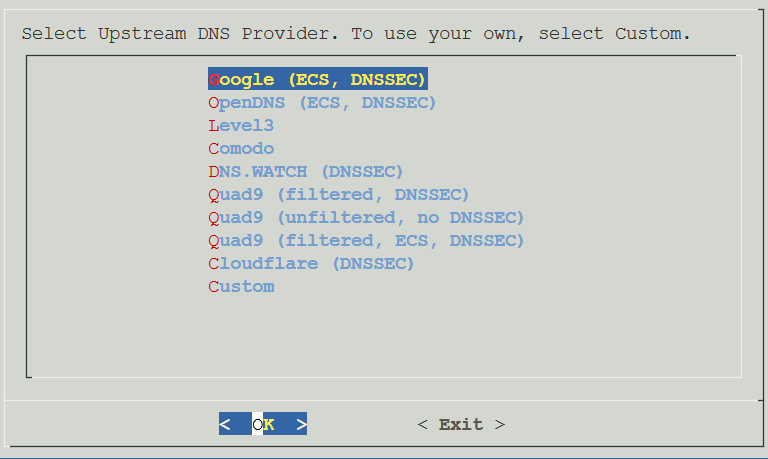
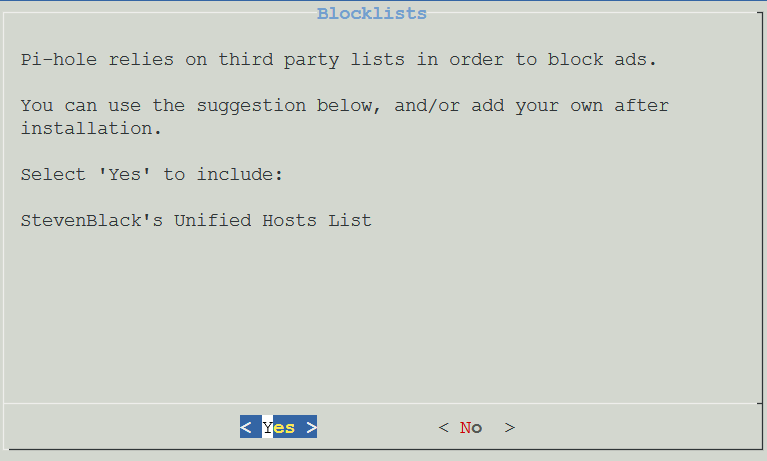
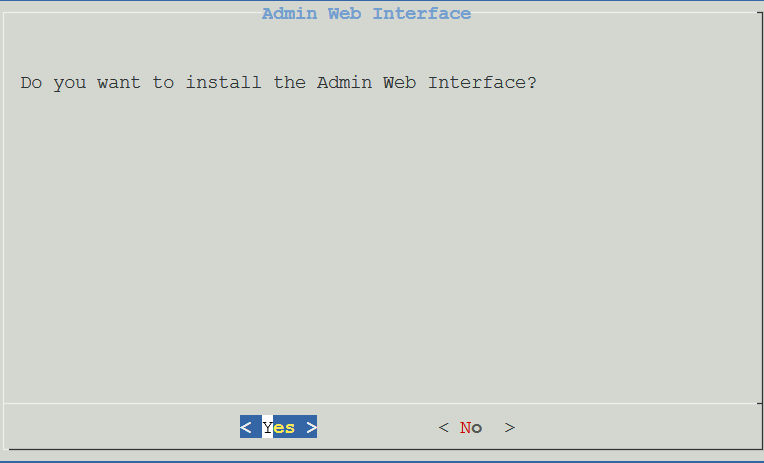
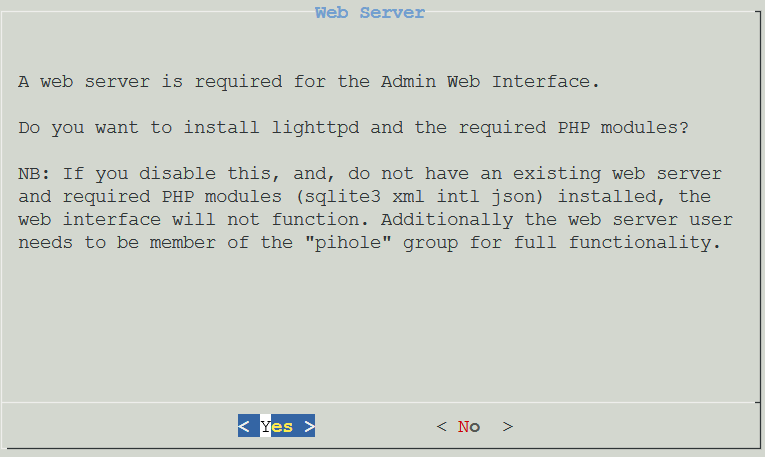
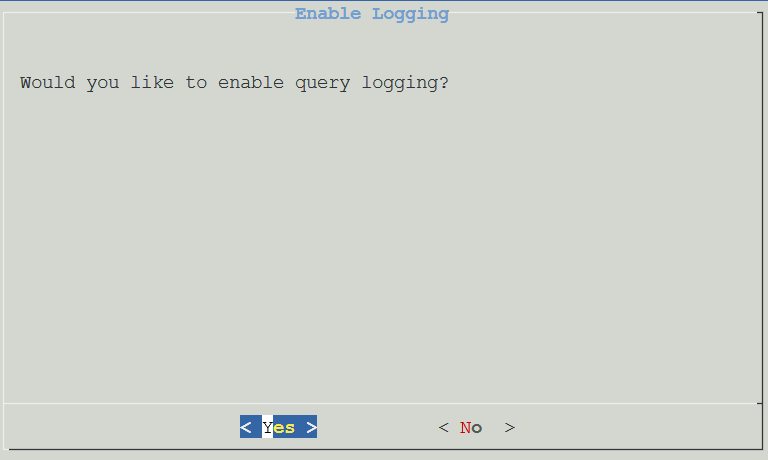
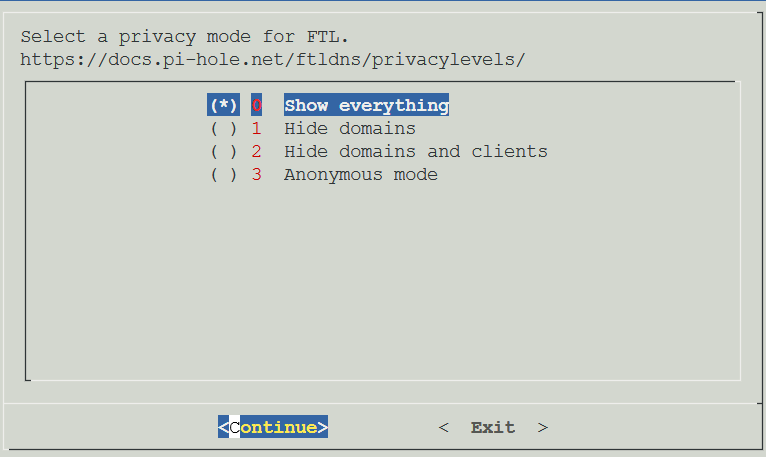
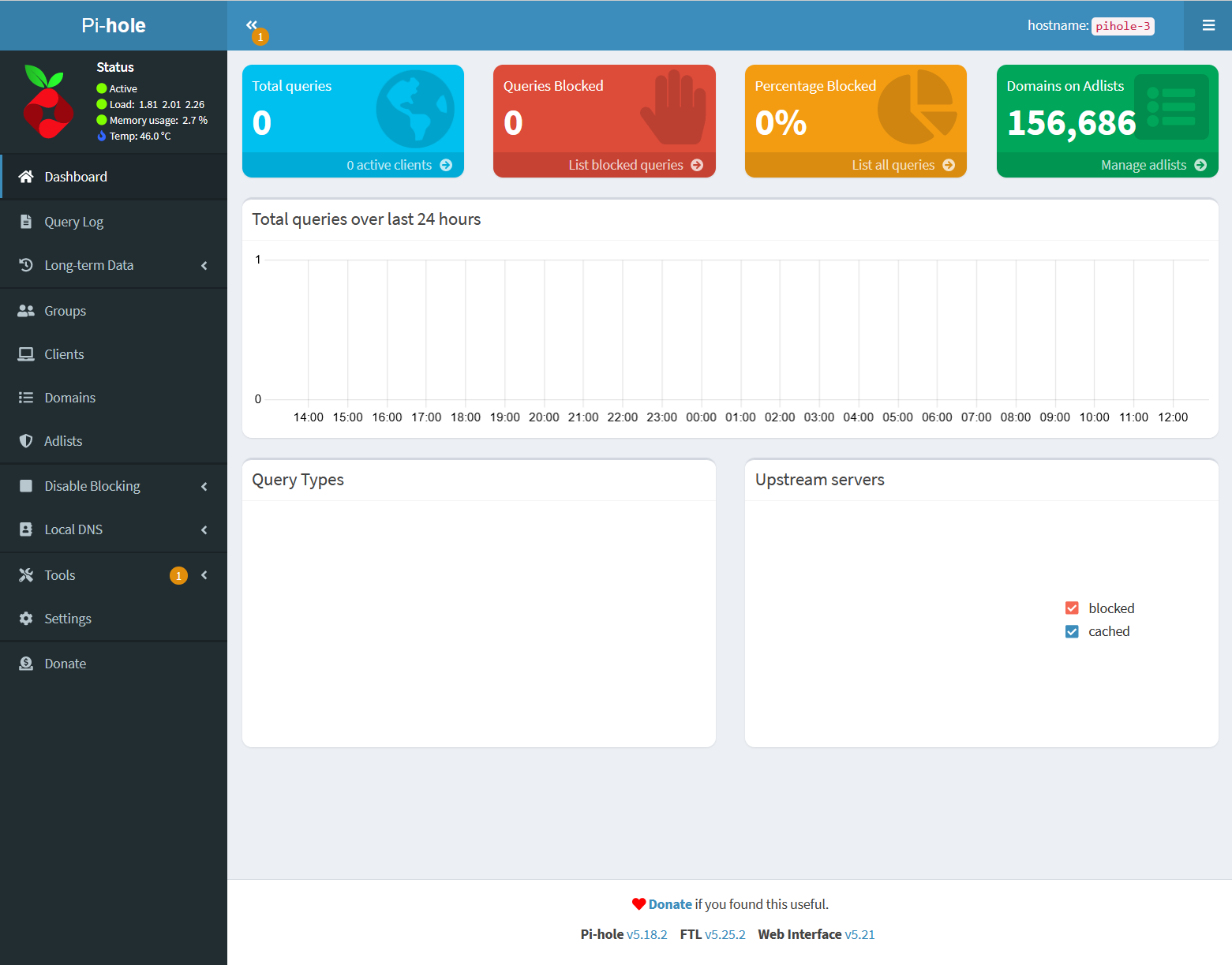
1. Curl is used to run script that installs docker.
2. Enter:  
     
   **reboot**

# 2. Setting up and Configuring Pi-hole:

1. Following on from the previous step, run the following script:  
     
   **curl -sSL https://install.pi-hole.net | bash**  
     
   [ side Note:  
    as of 31st March 2024 this gets:  
   Pihole: v5.18.2  
   FTL: v5.25.2  
   Web Interface: v5.21  
     
   May need to install the above versions somehow in the future if in later versions my use of ‘custom.list’ later on stops working  
   ]

This will quickly appear and may also disappear quickly:



1. Press enter until you get to this screen:  
   
2. Press TAB to select continue and press Enter
3. On this screen, ensure ‘Google (ECS, DNSSEC) is selected and OK is selected and press Enter:  
   
4. On the Blocklists screen, select Yes and press Enter:  
   
5. For Admin Web Interface, select Yes and press Enter:  
   
6. For the next screen, select Yes and press Enter:  
   
7. For the next screen, select Yes and press Enter:  
   
8. Select Show Everything, select Continue and press Enter  
   
9. Pi-hole will now install itself and finish by showing:  
   
10. Connect to Pihole at its admin endpoint, with the above admin password:  
    <http://192.168.124.158/admin>
11. You will be presented with:  
    
12. Logout of Pi-hole and log into the container console to change password, following these steps:  
      
    The Web interface password needs to be reset via the command line on your Pi-hole.

Use the pihole command to do this:

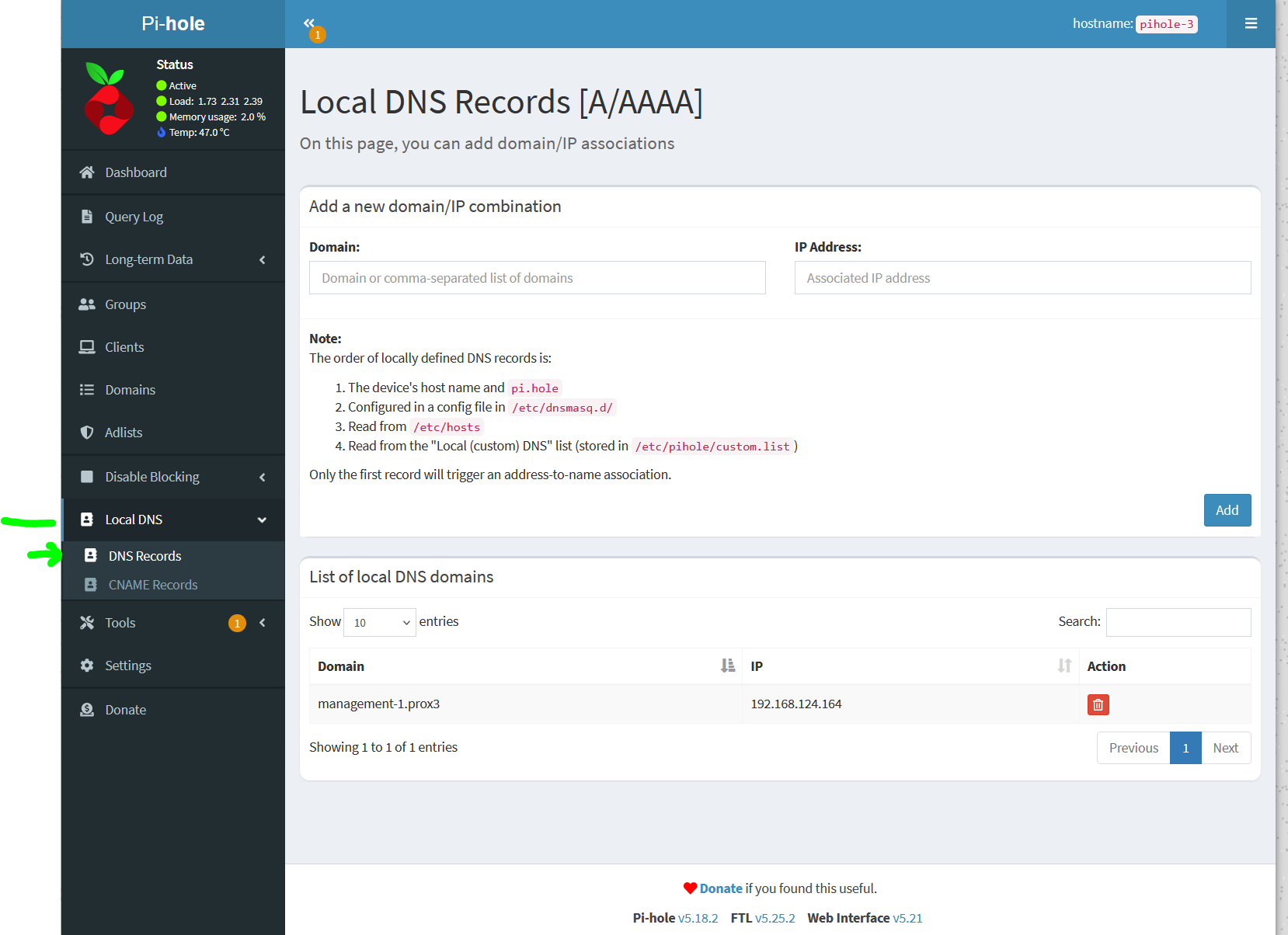
**pihole -a -p**

You will be prompted for the new password.

If you enter an empty password, the password requirement will be removed from the web interface.

1. That completes initial setup of Pi-hole.

# 3. Adding Local DNS Records for VM’s in Homelab:

1. In Pihole, go to this screen ‘Local DNS’:  
   
2. In the top section ‘Add a new domain/IP combination’ enter a homelab VM Domain name and Ip Address as per the one set up as seen in the lower section ‘List of local DNS domains above.
3. To test a VM can see / use Pi-hole, change its **/etc/netplan/50-cloud-init.yaml** file such that its **nameservers** section is as below (you’ll need to be root for this):  
     
   network:

version: 2

ethernets:

eth0:

addresses:

- 192.168.124.180/24

gateway4: 192.168.124.1

match:

macaddress: 22:5a:f8:b3:f5:f5

nameservers:

addresses:

**- 192.168.124.158**

search: []

set-name: eth0

1. Save the change
2. Also edit **/etc/systemd/resolved.conf** and change the line:  
     
   **Domains=192.168.124.162**  
     
   to:  
   **Domains=192.168.124.158**
3. Save the change and reboot the VM
4. Log back into the changed VM and if all is well, the following should work:  
     
   ping management-1.prox3  
     
   This achieves a simple test to see that things are working.
5. For other needed local DNS, log into the pihole VM as root and make the changes in the file:  
   **/etc/pihole/custom.list**   
   For homelab we don’t want domain names of format ‘**management-1.p3**’ because ‘**pki**’ is not playing ball with a domain name of ‘**pki.p3**’ … but it is happy with just ‘**pki**’ …  
   So replace the one line in **custom.list** with:  
     
   **192.168.124.164 management-1**

**192.168.124.164 pki**

**192.168.124.165 concourseworker-1**

**192.168.124.166 concourse-web**

**192.168.124.167 bastion-1**

**192.168.124.168 web-1**

**192.168.124.169 webasg-1**

**192.168.124.170 consul-1**

**192.168.124.170 nomad-1**

**192.168.124.171 consul-2**

**192.168.124.171 nomad-2**

**192.168.124.172 consul-3**

**192.168.124.172 nomad-3**

**192.168.124.173 vault-1**

**192.168.124.174 vault-2**

**192.168.124.175 vault-3**

**192.168.124.176 concourse-psql**

**192.168.124.177 gitea-1**

**192.168.124.178 logstash-1**

**192.168.124.179 elastic-1**

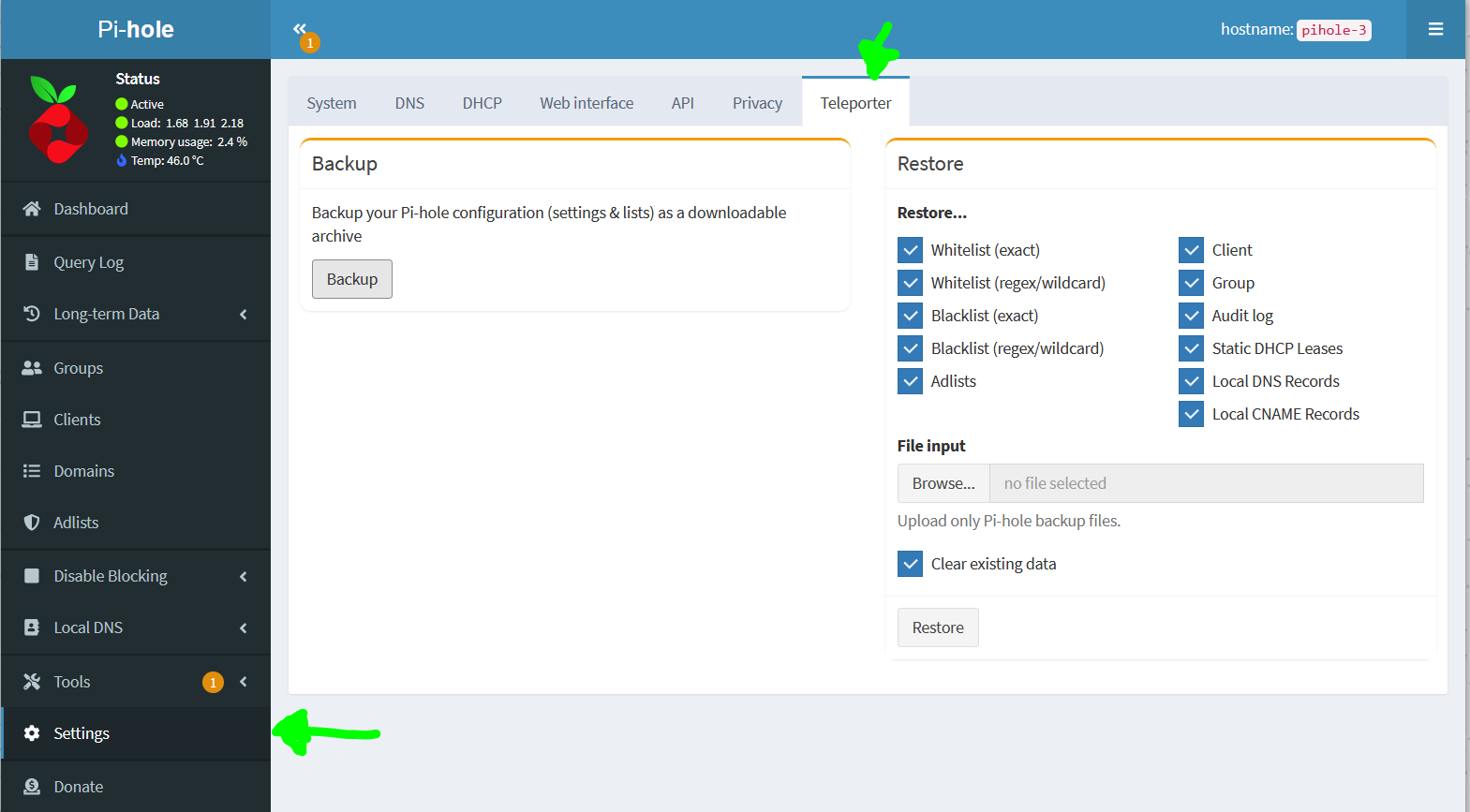
**192.168.124.180 kibana-1**

**192.168.124.181 prometheus-1**

**192.168.124.182 grafana-psql**

**192.168.124.183 webasg-2**

**192.168.124.184 webasg-3**  
  
and save.  
(Note: Pihole won’t allow the entry of a domain name as a single word in the user interface, it only allows the format: example.com … but editing the file gets something that seems to work for homelab)  
Wait a few seconds and reboot Pihole VM (or try as root: **service pihole-FTL restart** ) to reload the changes.  
Then test ping the edited domain name from a VM to check all is well.

1. Take a backup of the new settings from Pihole in Settings/Teleporter as per:  
   
2. Apply the above nameservers change to **/etc/netplan/50-cloud-init.yaml** in all needed VM’s, together with the change to **/etc/system/resolved.conf**
3. Once all VM’s are updated, remove any old hostnames from **/etc/hosts** file for the VM’s whose **/etc/netplan/50-cloud-init.yaml** file has been changed.  
   Reboot and check the removed hostnames have not reappeared in **/etc/hosts** file … if they have then remove any old hostnames from: **/etc/cloud/templates/hosts.debian.tmpl**
4. For all VM’s in proxmox, in Options adjust their start order such that pihole-3 is order 1 with ‘up 15’, run3 is order 2 with ‘up 30’ and others have their order bumped by one … and adjust any other homelab setup files with this new information.
5. Adjust run3 host to use Pi-hole as a secondary DNS and if ok, clean up its hosts (and possibly: **hosts.debian.tmpl)** file(s) and update any document for run3 host.  
   Will also need on run3 to do the following:  
      
   **apt install resolvconf**

**systemctl status resolvconf.service**

**nano /etc/resolvconf/resolv.conf.d/head**  
  
and add these lines:  
**nameserver 192.168.124.158**

**nameserver 8.8.8.8**

**nameserver 8.8.4.4**

then run:

**resolvconf --enable-updates**

**resolvconf -u**

**systemctl restart resolvconf.service**

**systemctl restart systemd-resolved.service**

**resolvectl status**

1. Unfortunately resolving still did not play ball on the VM’s other that run3 and I had to update my ansible role/resolver to achieve the steps in step 13 above.
2. That’s all.