Reztek Systems

QNAP Container Station

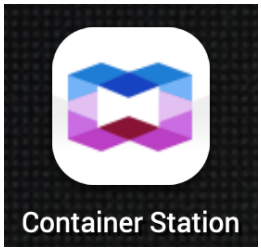
PiHole Setup Procedure

Last Updated: 2019/10/30

Prerequisites:

1. QNAP NAS running QTS 4.4.1
2. Completed installation of QNAP Container Station using App Center
3. Application of all updates for QTS and the installed Container Station program
4. Network configuration established for the planned deployment methodology:
    a. Assignment of the desired Static IP address for Bridging

Within the QNAP web interface, open the **Container Station** application.



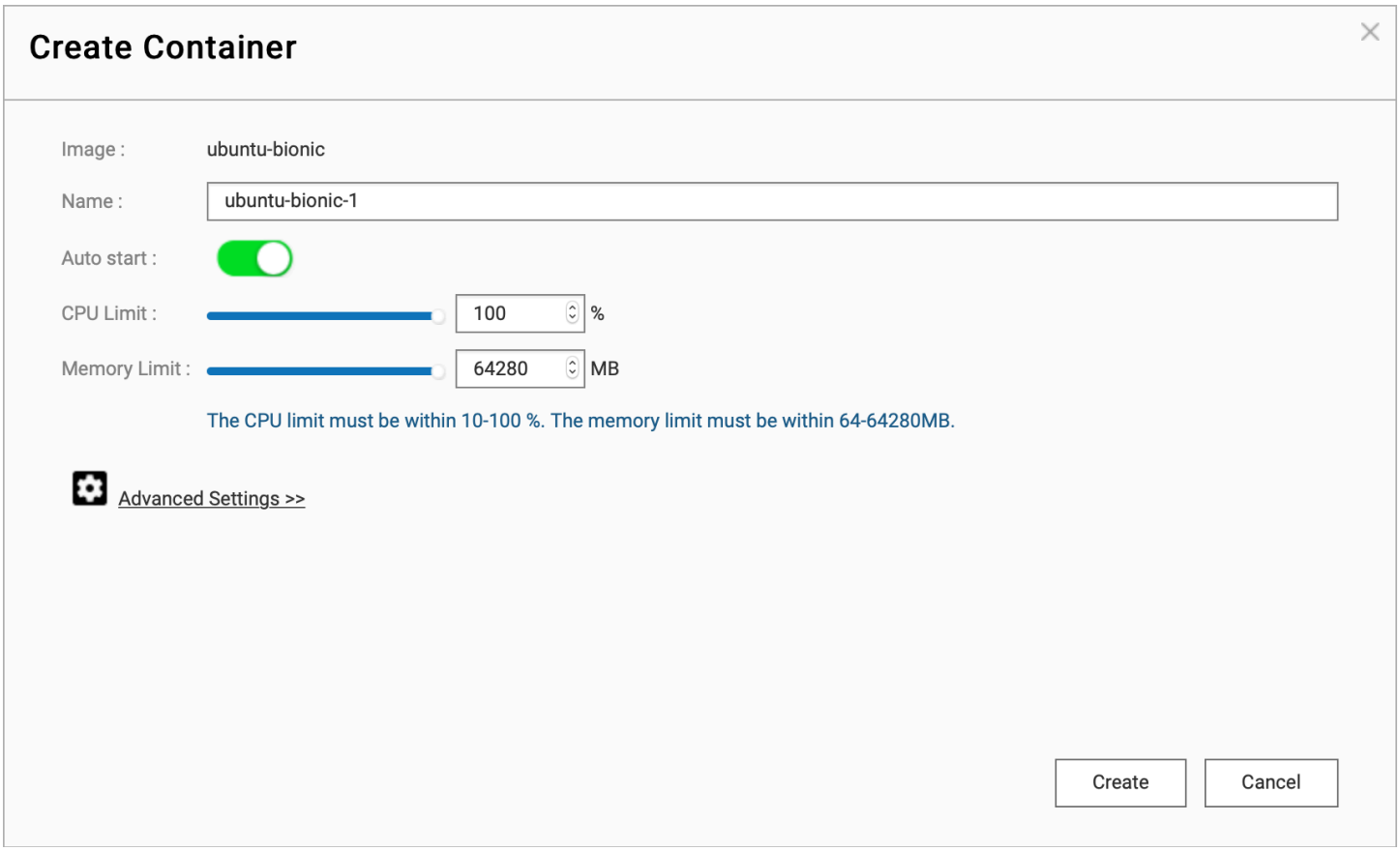
Once the interface has loaded, select the **Create** option under the **Management** menu on the left-hand side of the screen.

NOTE: The various Docker containers which may be returned by searching for **pihole** or **pi-hole** will not be used due to errata within the Container Station GUI. Attempts to install the vendor-provided pihole containers using the GUI will result in errors and an inability to properly start up the container. More information can be found in the following QNAP Support Forum thread: https://forum.qnap.com/viewtopic.php?t=143406. If and when QNAP decides to fix the errata, this procedure will be updated.

Underneath the **Container** heading in the right-hand side of the screen, a number of Linux distributions which have been packaged in the Linux Container (LXC) format will appear. Based on the available offerings in the catalog, we'll be selecting Version 18.04 of the Ubuntu LXC option. If you prefer an alternative distribution, such as Ubuntu 16.04 or Fedora 28, this may be selected instead of the option which is selected within this procedure.

NOTE: Officially supported Linux distributions are documented within the Pi-Hole Documentation. Our selection and recommendation of Ubuntu 18.04

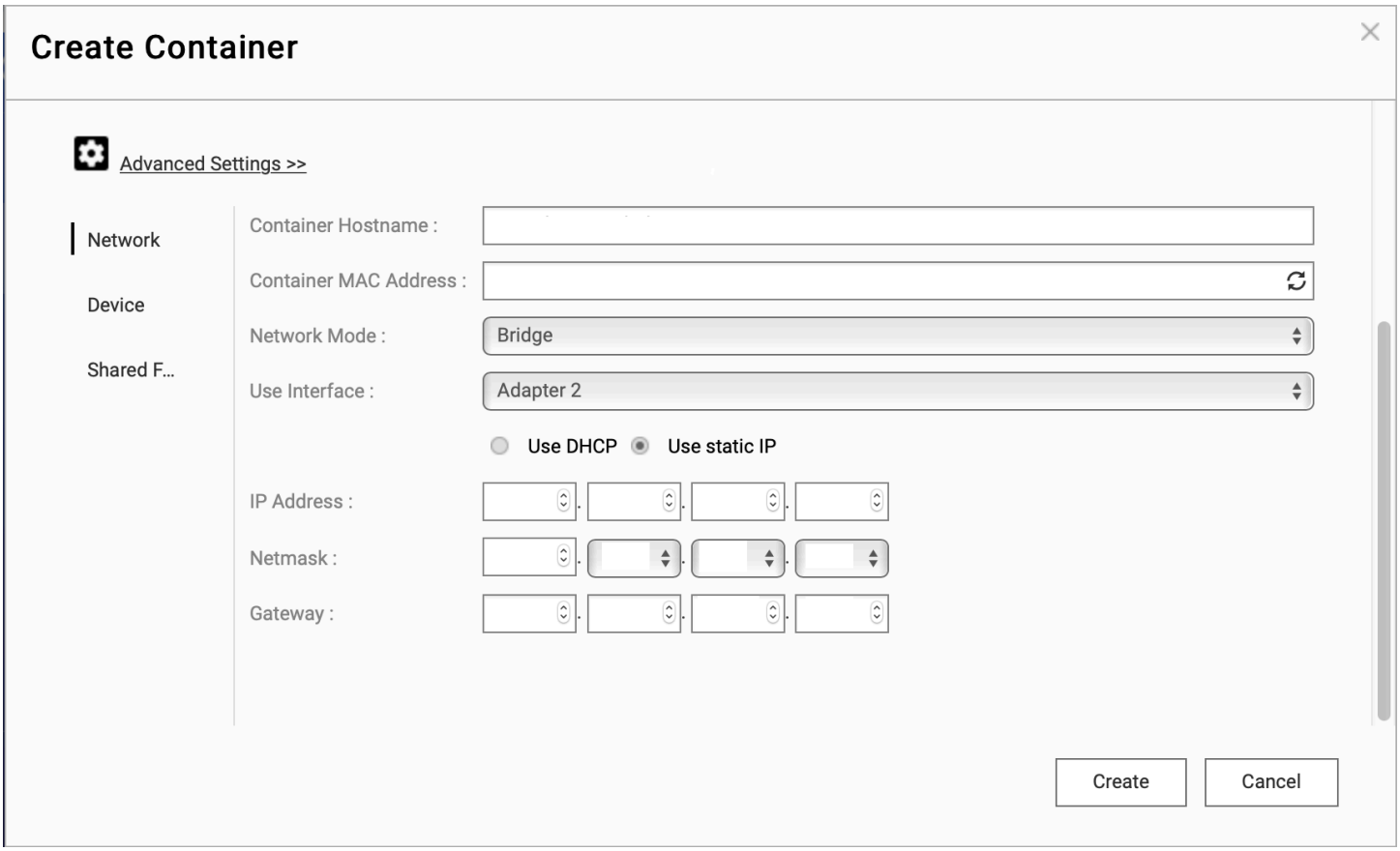
Left click the **Install** button underneath the preferred LXC version of choice. The **Create Container** pop-up window will appear.



The default maximum allocations for memory limit may be less than what is pictured based upon the QNAP NAS being used. For this container, adjust the following values:

- **Name:** Give it a meaningful name that aligns with whatever standards you may have in place. In the absence of a defined naming convention, utilize a name which will allow you to easily identify the purpose of the container (i.e. pihole).
- **CPU Limit:** 10%-20%. In the scope of a home network with a limited number of devices, the 10% limit will be more than sufficient when using a recent x86-based QNAP NAS. For larger deployments or for ARM-based QNAP NAS units, this limit may need to be increased to 20% or more.
- **Memory Limit:** 512 MB-1024MB. 512MB is the minimum requirement noted within the PiHole documentation and has been the default configuration we've used successfully. For NAS units which support 16GB or more of memory, doubling this allocation is feasible.

Once adjustments have been made to the values for Name, CPU Limit, and Memory Limit, additional options may be configured under Advanced Settings to establish the network configuration. **Left click** on **Advanced Settings** near the bottom of the window to bring up the configuration options for networking.



Within the Network stings, populate the following fields accordingly:

- **Container Hostname:** This should be set to match the Name assigned to the container in the prior step.
- **Network Mode:** Change this setting to **Bridge** to bind the connection to a physical port on the QNAP NAS. In our example, we're using the integrated Adapter 2. If your NAS only contains a single network port, you'll either need to use the sole interface or may leverage **NAT** instead.
- **Use DHCP or Use Static IP:** The PiHole container will require a static IP address. Select the **Use static IP** radio button and assign a valid and available IP address from your network.

The **Device** and **Shared Folders** options under **Advanced Settings** allows you to map existing shared folders or expansion cards to the container. Since the base PiHole functionality does not necessarily require either of these potential configuration options, **left-clicking** the **Create** button will allow you to proceed to the review portion of this process.

A summary window will appear with all of the customizations we've just discussed. Confirm that the resource allocations and network configuration by reviewing the contents of this final screen. If everything is set as desired, **left click** the **OK** button.

Near the top of the Container Station window, a Background Task which shows the progress of the container creation will appear. **Left click** the icon on the screen (pictured below) to monitor the progress of the creation task.

Once the deployment has completed, **left click** the **Containers** (under **Resource**) on the left-hand side of the screen. You will see the container with the assigned name in the list on the right-hand side of the screen. **Left click** on the container name to gain access to the console. Once you see the console prompt, we'd recommend changing the default password for the *ubuntu* user.

1.) Type **passwd** into the console and press **Enter**.
2.) Type **ubuntu** when prompted for the **(current) UNIX Password** and press **Enter**.
3.) Type in a new, strong password when prompted to **Enter new UNIX password** and press **Enter**.
4.) Retype the new, strong password when prompted to **Retype new UNIX password** and press **Enter**.

After a successful password change, the container will need to be updated.

NOTE: All commands involving apt may also be executed using apt-get.

1.) Type **sudo apt update** to check the repositories for available updates. You will be prompted to enter the new password you've established for the *ubuntu* user account.
2.) Once the update completes and a list of available packages appears, type **sudo apt upgrade** to download and apply the necessary updates.

Next, we'll create a dedicated user account which will be used for managing the PiHole application. You may name this account whatever you like. In the console (which should still be open as part of this process), type **sudo adduser *<user id of your choice>*** and press **Enter**. If prompted for the *ubuntu* account password, type it in and press **Enter**. Enter a unique and strong password which is _not_ the same password as the *ubuntu* account for this new user account. Additional prompts which appear for **Full Name**, **Room Number**, **Work Phone**, **Home Phone** and **Other** can be left blank. Type **Y** when the prompt to confirm that the information is correct appears and press **Enter**. This will create the new user account that will be used for the balance of processes. We'll also need to modify this account to place it in the desired group. Type **sudo usermod -aG sudo *<user id of your choice>*** and press **Enter**.

In order to administer and connect to the container outside of the console within Container Station, openssh-server will need to be installed. Type **sudo apt install -y openssh-server** into the console and press **Enter**. Once the install completes, you may either log the *ubuntu* user account out of the console within Container Station and log in using the newly created user account _or_ you may SSH to the assigned IP using PuTTY, Terminal or other method of choice.

NOTE: When connecting via SSH from outside of Container Station, you may receive a prompt related to the key fingerprint. Entering **yes** and pressing the **Enter** key will allow you to proceed with entering the account's password and establish the connection.

With remote access capability having been established, the final steps to bring PiHole to life will include:

1.) Installing curl
   a. Type **sudo apt install -y curl** within the console or SSH session and press **Enter**.
      i. A list of dependencies will also be presented. Type **Y** to confirm the install and press **Enter**.
   b. Type **sudo apt install -y resolvconf** within the console or SSH session and press **Enter**.
      i. Issues have been noted in the QNAP Support Forums related to the PiHole install within the container hanging at 79% where the resolvconf configuration occurs. Pre-installing this package will prevent the error from occurring. When prompted to keep the current version of */etc/resolvconf/resolv.conf.d/head,* accept the default of N by pressing **Enter** when prompted.
   c. Type **sudo apt install -y nano** within the console or SSH session and press **Enter**.
      i. Post-install processes for PiHole will require editing configuration files. The lightweight nature of the LXC does not contain nano. If you prefer a different text editor (i.e. vi, vim, emac, etc.), it may be called in place of nano. The keyboard shortcuts within nano are very similar to those within Microsoft Windows and will be more familiar for comfortable use.
   d. Once curl is installed, pihole will be installed by typing the following command into the console or SSH session followed by pressing **Enter**: **curl -sSL https://install.pi-hole.net | bash**
      i. Press Enter to accept the three prompts that appear related to the function of PiHole and the need for a static IP address.
      ii. From the Upstream DNS provider, select your preferred option. Selection will involve personal preference as well as potential privacy implications. We strongly encourage
      iii. From the third-party list perspective, keep all options enabled to address as many ads as possible. Press **Tab** until **Ok** is highlighted. Press **Enter**.

iv. Keep the options for IPv4 and IPv6 enabled. Press **Tab** until **Ok** is highlighted. Press **Enter**.
v. The static network interface which was previously defined for the container will appear in the **Static IP Address** window in CIDR notation. Confirm that the IP is correct. Press **Enter**.
vi. Install the web admin interface. The default option is **On (Recommended)**. Press **Tab** until **Ok** is highlighted. Press **Enter**.
vii. Install the web server (lighttpd). The default option is **On (Recommended)**. Press **Tab** until **Ok** is highlighted. Press **Enter**.
viii. Accept the default to log queries. Press **Tab** until **Ok** is highlighted. Press **Enter**.
ix. The privacy mode options will appear for FTL. **0 Show Everything** will provide an extensive level of detail related to client communications and DNS requests. Each subsequent option heading toward Disabled statistics will reduce logging and visibility into the associated request operations. The selection of the best option here comes down to personal preference in a home environment. Further details can be found on PiHole's site (https://docs.pi-hole.net/ftldns/privacylevels/). Select your preferred privacy mode and press **Tab** until **Ok** is highlighted. Press **Enter**.
x. The install routine will commence and required packages will be downloaded and applied. Once the install completes, a password for the Admin Webpage login will be displayed. Document this for initial login to the web interface before pressing **Enter**.

Validation that this container is functioning may be achieved by vising the Admin Webpage. Within a web browser, type the static IP address followed by **/admin**. An example would be *192.168.100.25/admin*. If the web interface appears, PiHole is operational. In order to use PiHole seamlessly, this IP address will need to be set as the Primary DNS server on your router, modem or other device which handles DHCP. Once that change is made, renewing DHCP leases via disconnecting and reconnecting from the associated network will refresh the DNS configuration and begin leveraging PiHole.

Automating the application of Ubuntu updates for this container may be achieved using the unattended-upgrades package. Within the console or SSH session, perform the following steps.

1.) Type **sudo apt -y install unattended-upgrades** and press **Enter**.
2.) Once this process completes, the configuration file will need to be modified. Type **sudo nano /etc/apt/apt.conf.d/50unattended-upgrades** and press **Enter**. Once the file loads in the console, the following modifications will need to be made.
    a. Remove the comments, indicated as **//**, in front of **"${distro_id}:${distro_codename}-updates";** by pressing the delete key twice.
    b. Scroll down until you find the following line:

i. **//Unattended-Upgrade::Remove-Unused-Kernel-Packages "false";**
   1. Remove the comments in front of this line by pressing the delete key twice to eliminate the **//**
   2. Change the value in between the quotes from **false** to **true**

c. Scroll down until you find the following line:
   i. **//Unattended-Upgrade::Remove-Unused-Dependencies "false";**
      1. Remove the comments in front of this line by pressing the delete key twice to eliminate the **//**
      2. Change the value in between the quotes from **false** to **true**

d. Scroll down until you find the following line:
   i. **//Unattended-Upgrade::Automatic-Reboot "false";**
      1. Remove the comments in front of this line by pressing the delete key twice to eliminate the **//**
      2. Change the value in between the quotes from **false** to **true**

e. Scroll down until you find the following line:
   i. **//Unattended-Upgrade::Automatic-Reboot-Time "02:00";**
      1. Remove the comments in front of this line by pressing the delete key twice to eliminate the **//**
      2. Change the value in between the quotes to a non-disruptive window where the container may automatically reboot (if 2:00 a.m. is not acceptable). Otherwise, leave this value intact.

Once these modifications have been made, save the file (press **Ctrl-O** in nano and press **Enter**). After the configuration file is saved, exit the editor (press **Ctrl-X** in nano). The enablement of automatic updates will require editing one more configuration file. Within the console or SSH session, perform the following steps.

1.) Type **sudo nano /etc/apt/apt.conf.d/20auto-upgrades** and press **Enter**. Once the file loads in the console, the following modifications will need to be made.
   a. The configuration file will contain items for **Update-Package-Lists** and **Unattended-Upgrade** with values set to **"1"**. Move the cursor to the next line.
      i. Type **APT::Periodic::Download-Upgradeable-Packages "1";** and press **Enter**.
      ii. Type **APT::Periodic::AutocleanInterval "7";** and press **Enter**.
      iii. Press **Ctrl-O** in nano and press **Enter** to save the updated configuration file.
      iv. Press **Ctrl-X** to quit nano.

Upon completion of these efforts, the container will automatically update daily, reboot if required by the update, and perform housekeeping for packages on a weekly basis.

References and Acknowledgements

The establishment and refinement of this process includes knowledge and information provided by the following sources:

Vmsman in the QNAP Forums for development of the Ubuntu 16.04 LXC process: https://forum.qnap.com/viewtopic.php?f=354&t=144282
DIY_Glenn in the QNAP Forums for the resolvconf fix: https://forum.qnap.com/viewtopic.php?t=147501&start=15
Libre-software.net for the automatic maintenance process: https://libre-software.net/ubuntu-automatic-updates/