# Pi-Ager install on Raspberry Pi 3/Pi 4/Pi zero (2)w under Pi OS 12 Lite bookworm

* For all Pi:

Download and install Raspberry Pi Imager v1.8.4 or later.

Start Raspberry Pi Imager, then select:

Raspberry Pi Device:

NO FILTERING

Operating System:

Raspberry Pi OS Lite (32-bit) ( A port of Debian Bookworm with no desktop environment)

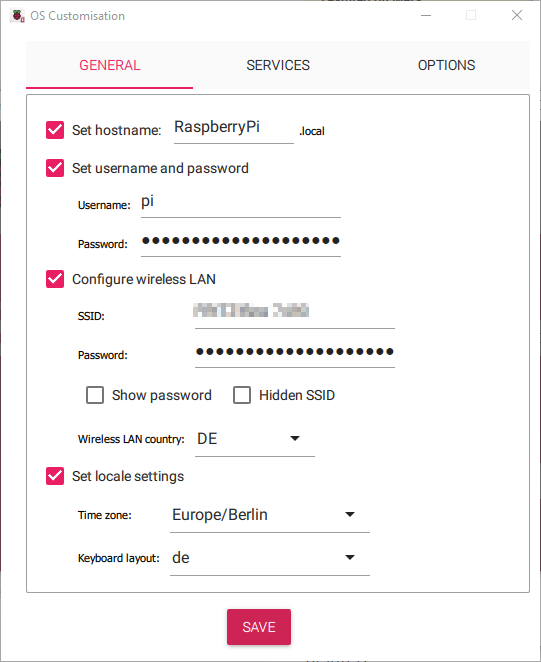
Storage:

Use an USB Reader and choose your SD-Card, min. 8 GB

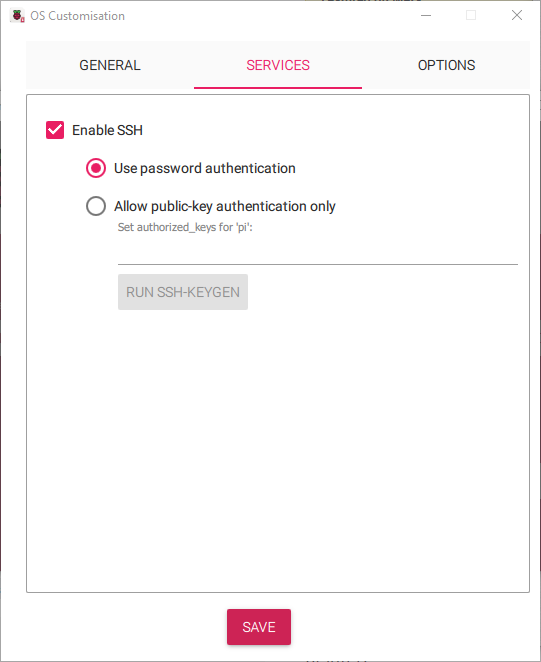
Then click ‘NEXT’

In popup Window ‘Use OS customisation’ select ‘EDIT SETTINGS’.

In the ‘GENERAL’ Tab edit your settings:



In the ‘SERVICES’ Tab enable SSH:



Then ‘SAVE’ and apply OS customisation settings with ‘YES’.

The customized OS is now written to your SD-Card.

Put your SD-Card in your Raspberry Pi and power on your Pi device.

Login via SSH (e.g. use PuTTY) or connect a HDMI monitor and USB keyboard to

your Raspberry device and continue with the setup as described below.

* To allow root login:

In /etc/ssh/sshd\_config : remove # from PermitRootLogin and replace prohibit-password to yes. Then restart ssh server: sudo service ssh restart

Set your root password with: sudo passwd root

* Run raspi-config and disable login shell on serial port and enable serial port
* Edit config.txt in /boot/firmware to support I2C and SPI devices:

CAUTION: If you want to edit config.txt ,cmdline.txt and setup.txt under Windows in a Terminal Window, e.g. using Terminal Apps like PuTTY, the Linux folder /boot/firmware on the SD-Card is mapped to a windows partition FAT32 USB Drive named bootfs and contains the above mentioned files.

# Additional overlays and parameters are documented /boot/overlays/README

# Use Pi-Ager Pins 11/13 GPIO 17/27 for I2C

dtoverlay=i2c-gpio,bus=3,i2c\_gpio\_sda=17,i2c\_gpio\_scl=27

# Use Pi-Ager Pin 16 for MCP3204

dtoverlay=spi1-1cs,cs0\_pin=16

at the end of config.txt add the following lines to support bluetooth and Nextion TFT displays

via serial port /dev/serial0 :

[all]

enable\_uart=1

dtoverlay=miniuart-bt

# force\_turbo=1

* Add in /boot/firmware/cmdline.txt at the end of line this to enable USB camera with fswebcam :

dwc\_otg.fiq\_fsm\_mask=0x3

• Reboot system

* Edit /etc/modules to load i2c-dev at boot, add this line :

i2c-dev

* Add file :

sudo touch /etc/modprobe.d/raspi-blacklist.conf

* Install git: sudo apt install git
* Get a copy from Pi-Ager repository to your local system:

git clone –depth=1 –b master <https://github.com/Tronje-the-Falconer/Pi-Ager>

All project file are now in the folder ./Pi-Ager/

Change working directory to Pi-Ager

cd Pi-Ager

* Copy setup.txt from local repository to /boot/ and edit it as needed:

sudo cp ./boot/setup.txt /boot/firmware/

Create a symbolic link in /boot to /boot/firmware/setup.txt

cd /boot

sudo ln –s /boot/firmware/setup.txt setup.txt

* Copy /etc/modprobe.d/Pi-Ager\_i2c\_off.conf.on from local repository to /etc/modprobe.d/

sudo cp ./etc/modprobe.d/Pi-Ager\_i2c\_off.conf.on /etc/modprobe.d/

* Reboot system
* Update system:

sudo apt-get update

sudo apt-get upgrade

• Install php 8:

sudo apt install php-fpm php-cli

sudo apt install php

sudo apt install php-common php-sqlite3

* Install additional modules for php:

sudo apt install php-mbstring php-zip php-curl

* Install lighttpd:

sudo apt-get install lighttpd

sudo systemctl status lighttpd

sudo nano /etc/lighttpd/lighttpd.conf

and change/add Parameter

server.document-root = "/var/www/html"

to

server.document-root = "/var/www"

add “mod\_auth” to server.modules list

add “mod\_authn\_file” to server.modules list

Save and close nano

To enable PHP8 in Lighttpd, we must modify /etc/php/8.2/fpm/php.ini and uncomment the line cgi.fix\_pathinfo=1. In my php.ini file, I found it on line 807. To uncomment, just remove the semicolon in the beginning.

sudo usermod -G www-data -a pi

sudo chown -R www-data:www-data /var/www

sudo chmod -R 755 /var/www

* Reboot system

For testing the web server, generate html-page:

sudo nano /var/www/test.html

with content:

<html>

<head><title>Test-Seite</title></head>

<body>

<h1>Das ist eine Test-Seite.</h1>

</body>

</html>

Change owner: sudo chown www-data:www-data /var/www/test.html

Enter your IP Address into the browser followed by /test.html

In addition we need .htcredentials to contain user and password.

For that we use the Online-Tool https://websistent.com/tools/htdigest-generator-tool/

Username: pi-ager

REALM: Pi-Ager

Password: raspberry

Caution! All entries are case sensitive!

Open this file now

sudo nano /var/.htcredentials

and fill in the string output from the generator tool.

Save file with “STRG+o”, “RETURN” and close with “STRG+x”

Now we have to setup the password authentification in lighttpd:

sudo nano /etc/lighttpd/conf-available/05-auth.conf

The following lines are added under server.modules += („mod\_auth“) :

auth.backend = "htdigest"

auth.backend.htdigest.userfile = "/var/.htcredentials"

auth.require = ( "/settings.php" =>

(

"method" => "digest",

"realm" => "Pi-Ager",

"require" => "user=pi-ager"

),

"/admin.php" =>

(

"method" => "digest",

"realm" => "Pi-Ager",

"require" => "valid-user"

),

"/webcam.php" =>

(

"method" => "digest",

"realm" => "Pi-Ager",

"require" => "valid-user"

),

"/notification.php" =>

(

"method" => "digest",

"realm" => "Pi-Ager",

"require" => "valid-user"

)

)

Then we activate this modul:

sudo lighty-enable-mod auth

In addition we have to edit :

sudo nano /etc/lighttpd/conf-available/15-fastcgi-php-fpm.conf

Add “broken-scriptfilename” and “x-sendfile” :

fastcgi.server += ( ".php" =>

((

"socket" => "/run/php/php-fpm.sock",

"broken-scriptfilename" => "enable",

"x-sendfile" => "enable"

))

)

save and close nano

Now enable these modules:

sudo lighty-enable-mod fastcgi-php-fpm

Now reload the the webserver:

sudo service lighttpd force-reload

Now continue to install additional modules:

* Install smbus

sudo apt-get install python3-smbus

* Install sqlite3:

sudo apt install sqlite3

* Install pip3

sudo apt install python3-pip

CAUTION : sudo pip3 install <module> no longer supported.

If you see an error message, try to install the module with

sudo apt install python3-<module name>. If this does not work, try install modules with

sudo pip3 install <module name> –break-system-packages

error: externally-managed-environment

× This environment is externally managed

╰─> To install Python packages system-wide, try apt install

python3-xyz, where xyz is the package you are trying to

install.

If you wish to install a non-Debian-packaged Python package,

create a virtual environment using python3 -m venv path/to/venv.

Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make

sure you have python3-full installed.

If you wish to install a non-Debian packaged Python application,

it may be easiest to use pipx install xyz, which will manage a

virtual environment for you. Make sure you have pipx installed.

See /usr/share/doc/python3.11/README.venv for more information.

note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can override this, at the risk of breaking your Python installation or OS, by passing --break-system-packages.

hint: See PEP 668 for the detailed specification.

* Install DHT sensor support

sudo pip3 install Adafruit-DHT

* Install SHT1x sensors

sudo pip3 install pi-sht1x

* Install libgd-dev (needed for new version of fswebcam)

sudo apt install libgd-dev

* Install openssl dev package

sudo apt install libssl-dev

* Install fswebcam:

sudo apt install fswebcam

* Install cryptography

sudo apt install python3-cryptography

* Install uuidgen

sudo apt install uuid-runtime

* Install wiringpi new version with Pi4 support :

cd /tmp

wget https://project-downloads.drogon.net/wiringpi-latest.deb

sudo dpkg -i wiringpi-latest.deb

* Copy gpio to /usr/local/bin

sudo cp /usr/bin/gpio /usr/local/bin

sudo chmod 4755 /usr/local/bin/gpio

* Install PiShrink ( not longer needed )

wget <https://raw.githubusercontent.com/Drewsif/PiShrink/master/pishrink.sh>

chmod +x pishrink.sh

sudo mv pishrink.sh /usr/local/bin

* Nextion serial client (HMI Dislplay support)

sudo pip3 install nextion

* Install lsof command:

sudo apt update

sudo apt install lsof

* Install Locale en-GB and de-DE UTF-8 using

sudo raspi-config

* Enable Serial Interface, disable login, needed for HMI Nextion Display

sudo raspi-config

* Install zip and unzip:

sudo apt install zip unzip

* Install schedule

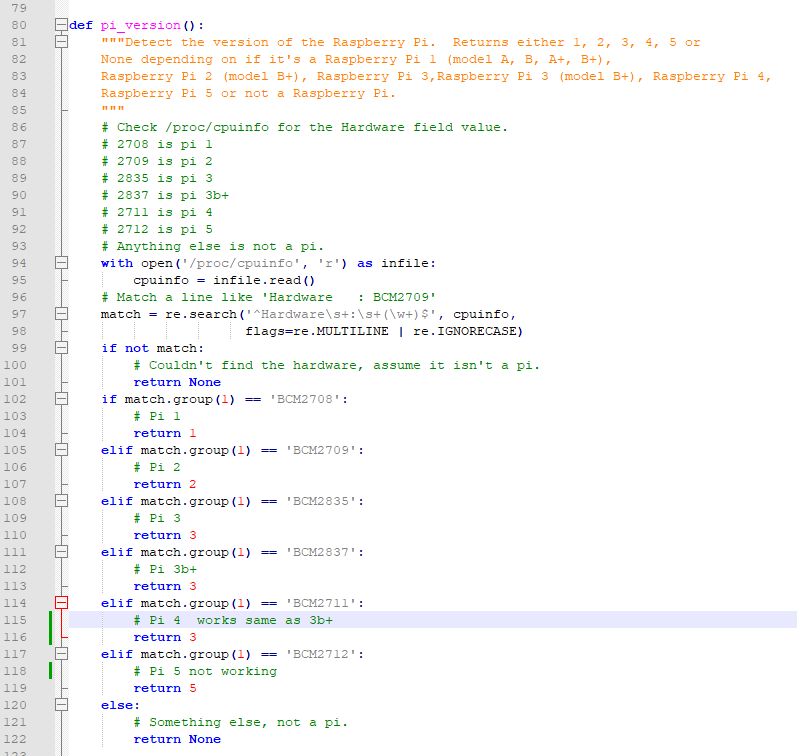
sudo pip3 install schedule

* Install MQTT client

sudo apt install python3-paho-mqtt

* Workaround for Adafruit\_DHT for Pi4:

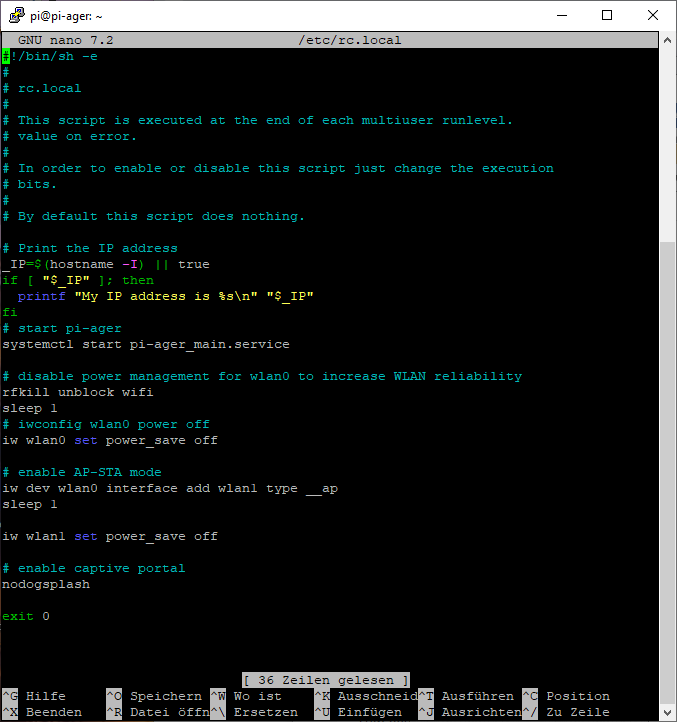
In "/usr/local/lib/python3.11/dist-packages/Adafruit\_DHT/platform\_detect.py", you can add/modify some lines, so it should workaround the issue, that GPIO Control does not work.



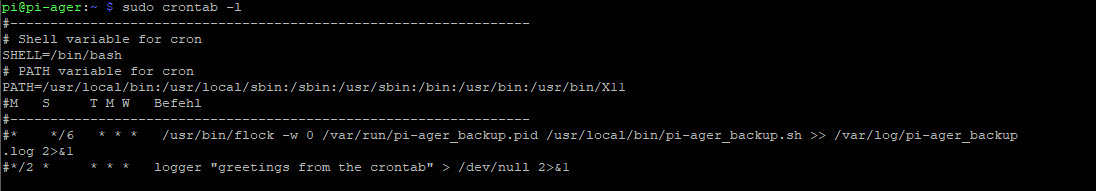
* Unblock wifi for Pi4, add rfkill unblock wifi and disable power management for wlan0:

cd /etc

sudo nano rc.local



* Only if wanted: Generate/edit crontab to prepare for automatic enable pi-ager\_backup.sh



* Use visudo to edit /etc/sudoers, so that the www-data User (User of Website) can execute /var/sudowebscript.sh :

sudo visudo

and then in sudoers following

...

#User privilege specification

root ALL=(ALL:ALL) ALL

...

adding:

www-data ALL=NOPASSWD:/var/sudowebscript.sh, /var/show\_wifi\_connections.sh, /var/updatessid.sh,/usr/bin/raspi-config,/usr/bin/nmcli

Save and exit.

* Install access point with network manager:

Copy from repository /usr/local/bin/setup\_pi-ager-ap.sh to destination.

Create virtual interface wlan1:

sudo iw dev wlan0 interface add wlan1 type \_\_ap

then run script :

cd /usr/local/bin

sudo ./setup\_pi-ager-ap.sh

* Install nodogsplash captive portal

sudo apt install iptables

sudo apt install libmicrohttpd-dev

cd ~

git clone <https://github.com/nodogsplash/nodogsplash.git>

cd ./nodogsplash

make

sudo make install

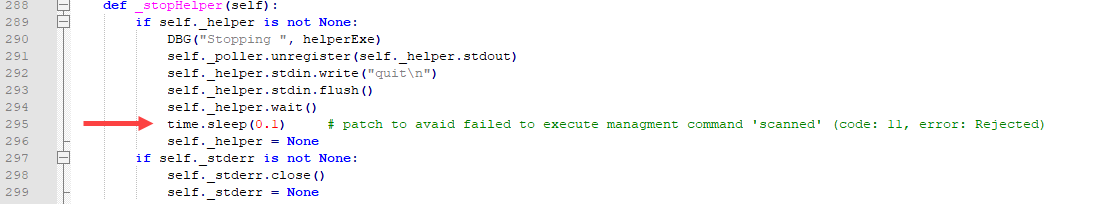
copy all from repository /etc/nodogsplash to /etc/nodogsplash

sudo chmod +x /usr/bin/nodogsplash

sudo chmod +x /usr/bin/ndsctl

* Install Bluetooth modules to support Bluetooth Temp./Hum Sensor from Xiaomi
* sudo apt install libglib2.0-dev
* sudo pip3 install bluepy requests
* sudo apt install bluetooth libbluetooth-dev
* # sudo pip3 install pybluez pycryptodomex
* Patch btle.py to stop intermittend errors:

File : /usr/local/lib/python3.11/dist-packages/bluepy/btle.py :



* Now copy all files and folders from your local git repository /var/www to /var/www/
* from local repository /opt/pi-ager/ to /opt/pi-ager/
* from local repository /var/sudowebscript.sh to /var/
* sudo chown –R www-data:www-data /var/www
* sudo usermod –G gpio –a www-data
* sudo chmod 666 /var/www/logs/logfile.txt
* sudo chmod 755 /var/www/logs/
* sudo chmod 664 /var/www/config/pi-ager.sqlite3
* sudo chown -R www-data:www-data /var/www/config/
* sudo chmod 777 /var/www/config/
* sudo chmod 555 /var/sudowebscript.sh /var/updatessid.sh /var/show\_wifi\_connections.sh
* from local repository /usr/local/bin/\*.sh copy all to /usr/local/bin/

(pi-ager\_backup.sh, pi-ager\_image.sh,setup\_pi-ager.sh)

Set +x mode to the scripts :

sudo chmod +x /usr/local/bin/\*.sh

* from local repository /lib/systemd/system copy the following files to

/lib/systemd/system/ :

pi-ager\_main.service

setup\_pi-ager.service

Then activate these services:

sudo systemctl daemon-reload

* from local repository /usr/bin copy the following file to /usr/bin/. This is a newer version of fswebcam with re-get frame on error.

fswebcam

Set +x mode to fswebcam:

sudo chmod +x /usr/bin/fswebcam

* from local repository /usr/share/man/man1/fswebcam.1.gz copy the following file to /usr/share/man/man1/

fswebcam.1.gz

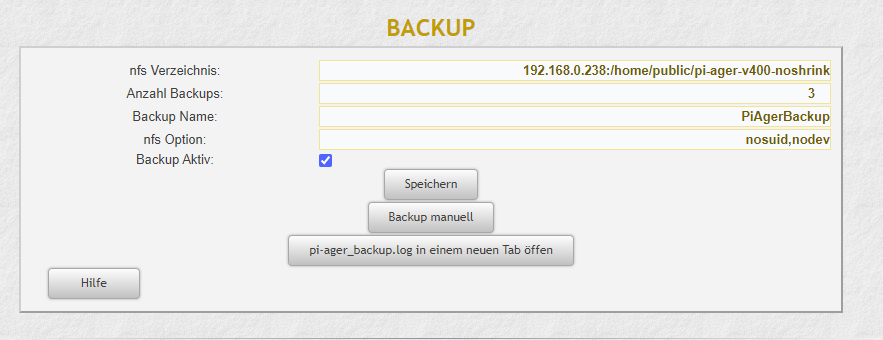
* Restart system

sudo reboot

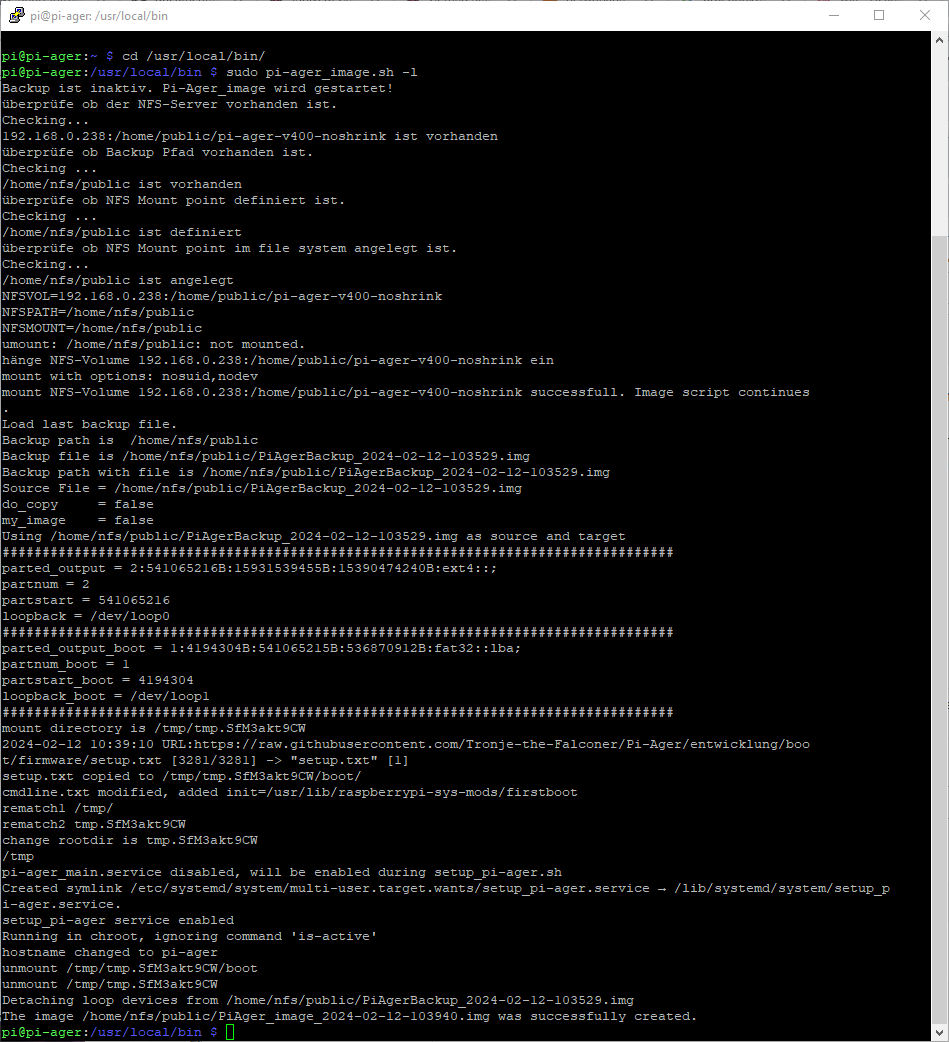
# How to generate and deploy Pi-Ager Images

There are some steps to do for generating and deploying a new image from a running Pi-Ager system:

1. Setup a NFS Server to store a new image. The best way is using a Raspi 4B or 5 equipped with an USB Memory stick with 128GB or more. Using RPi 5 it is now possible to add a NVME SSD so that there will be enough space for storing one or more images.
2. On the NFS Server System Install Pi Power Tools using the well-known Pi-Apps.
3. On the Pi-Ager setup and start an Image backup on the ADMIN page. For example :



1. Next start a terminal on the Pi-Ager, e.g. using PuTTY, and start the pi-ager\_image.sh script with option -l:



1. On the NFS Server System open the folder where the new image resides.
2. Start the Pi Power Tools and select IMG Mode
3. Drag and drop the new generated Image from the folder to the input line underneath ‘Select an img to continue’
4. Click ‘Next’ to shrink the new Image to the minimum possible size
5. Zip and deploy your new image.