ryo-iso

Release 0.1.7

Jul 25, 2023

Contents:

1 ryo-iso **1**
- 1.1 Overview . . . 1
- 1.2 Usage . . . 2
- 1.3 Requirements . . . 2
- 1.4 Other commands . . . 2
- 1.5 License . . . 3

2 Configuration **5**
- 2.1 Project Configuration . . . 5
- 2.2 General Configuration . . . 8

3 Changelog **11**
- 3.1 [0.1.7] . . . 11
- 3.2 0.1.6 . . . 11
- 3.3 0.1.5 . . . 12
- 3.4 0.1.4 . . . 12
- 3.5 0.1.3 . . . 12
- 3.6 0.1.2 . . . 12
- 3.7 0.1.1 . . . 13
- 3.8 0.1.0 . . . 13

4 Indices and tables **15**

Index **17**

CHAPTER 1

ryo-iso

Table of Contents

- *Installation*
- *Usage*
- *License*

1.1 Overview

ryo-iso is distributed on PyPI and is available for Python 3.8+ on Linux.

```
$ python3 -m pip install -U --user git+https://git.sr.ht/~lucidone/ryo-iso
```

1.1.1 Documentation

Documentation is available at https://ryo-iso.readthedocs.io/

1.1.2 Deprecated

- Ubuntu 16.04 (Xenial)
- Python 3.6

1.1.3 Currently supported targets

- Ubuntu 18.04 (Bionic)
- Ubuntu 20.04 (Focal)
- Ubuntu 22.04 (Focal)

1.2 Usage

1.2.1 TL;DR

```
$ sudo apt install curl gpgv2 squashfs-tools xorriso apt-utils apt-cacher-ng qemu-
↪system-x86 isolinux
$ if [ $(lsb_release -cs) = "jammy" ]; then sudo apt install jq fdisk; fi
$ python3 -m pip install -U --user git+https://git.sr.ht/~lucidone/ryo-iso
$ mkdir ~/iso_test
$ cd ~/iso_test
$ ryo-iso init
$ ryo-iso build
$ ryo-iso start
```

1.3 Requirements

`$ sudo apt install curl gpgv2 squashfs-tools xorriso apt-utils apt-cacher-ng qemu-system-x86 isolinux` Working with `jammy` also requires `$ sudo apt install jq fdisk`

1.4 Other commands

1.4.1 Create a new project

`$ ryo-iso init`

This command will initialize a project with a default `iso.yml` configuration file in the current directory.

If this is the first time being run it will create the `ryo-iso` application config file in `~/.config/ryo-iso/config.yml` and provides a set of reasonable defaults in `~/.config/ryo-iso/iso_base.yml` that can be overridden on a per-project basis.

1.4.2 Build an ISO

`$ ryo-iso build`

Builds an iso in `build/image.iso`

1.4.3 VM Install

`$ ryo-iso install`

Generated images can be tested by installing them into a QEMU VM

1.4.4 VM Start

```
$ ryo-iso start
```

This command can provide a means of booting the image as a LiveCD or restarting a previously installed disk image.

1.4.5 Cleanup

```
$ ryo-iso clean
```

This command will remove all build artifacts to prepare the project to be checked into version control.

NOTE: Using this command is preferable to running `rm -rf` as builds that are aborted with Ctrl-C may leave chrooted filesystems mounted.

1.4.6 Additional

```
$ ryo-iso list -p
```

This will list all intermediate processes that may be useful for debugging.

1.4.7 GPG Keys

All of the keys found in `./keys` will be install into the squashfs-root On `jammy` and later they should be referenced via `deb [arch=amd64 signed-by=/usr/share/keyrings/repo-keyring.gpg] http://repo.example.org/ ubuntu jammy main`

1.5 License

ryo-iso is distributed under the terms of both

- MIT License
- Apache License, Version 2.0

at your option.

CHAPTER 2

Configuration

2.1 Project Configuration

2.1.1 Overview

To initialize the config files and start a new project.

```
mkdir ~/iso_test
cd ~/iso_test
ryo-iso init
```

Note

By default, `ryo-iso` will load `iso.yml` from the current directory. This can be overridden by setting *RYO_ISO_CONF* to location of the config file.

2.1.2 Project Configuration File

Listing 1: iso.yml

```
# ryo-iso config file
###############################

# ISO Volume ID
name: RYO-ISO Example

# Base ISO image to rebuild
image: ubuntu/jammy

# Hardware Architecture
```

(continues on next page)

(continued from previous page)

```yaml
arch: amd64

# Base ISO Variant
#   ['desktop', 'live-server']
variant: desktop

# APT subtasks
apt:
  # add-apt-repository
  repository:
    - universe

  # apt-get install
  install:
    - git
    - python3-pip

# PIP subtasks
pip:
  # pip3 install
  install:
    - doit

# QEMU Settings
qemu:
  # VM disk size
  disk_size: 10G
  # Additional QEMU arguments
  args: "-device usb-host,vendorid=0x20a0,productid=0x4108"
```

Parameters

Parameter	Description
image	The *distro/version* of the upstream ISO image The version can be specified as the codename, or in yy.mm or yy.mm.patch for Ubuntu
arch	Target CPU architecture [‘amd64’,’i386’]
variant	Upstream ISO variant [‘desktop’, ‘live-server’]
name	Volume name of the generated ISO
apt[‘repository’]	List of apt repositories to add. Internally[1] this calls `add-apt-repository` within the chroot and supports PPAs and custom repositories.
apt[‘purge’]	List of packages to `apt-get purge`
apt[‘install’]	List of packages to install via apt
dpkg[‘install’]	List of packages install via `dpkg -i`
pip[‘install’]	List of packages installed via pip
pip[‘local’]	List of Python Wheels to install via pip *RYO_BUILD_DIR*/python-wheels/
patch	Script to be run for final local customizations
qemu[‘disk_size’]	Disk size for VM testing
qemu[‘usb-host’]	USB pass-through for VM testing

[1] See `ryo_iso.tasks.main.task__squashfs_apt_repo()`

Examples

Image

```
# Image by codename
image: ubuntu/jammy

# Image yy.mm
image: ubuntu/22.04

# Image yy.mm.patch
image: ubuntu/22.04.1
```

apt['repository']

```
apt:
  repository:
    # Copy sources.list from ${RYO_BUILD_DIR} into chrooted
    # /etc/apt/sourced.list
    # NOTE: This file is restored to the upstream version before building the ISO
    - sources.list
    # Copy ${RYO_BUILD_DIR}/sources.list.d/ros-latest.list into
    # /etc/apt/sources.list.d
    - sources.list.d/ros-latest.list
    # Install PPA and repository key
    - ppa:hxr-io/turtlebot

apt:
  repository:
    - universe
    - multiverse

apt:
  repository:
    - deb [arch=amd64 signed-by=/usr/share/keyrings/repo-keyring.gpg] http://repo.
↪example.org/ubuntu jammy main
    - deb-src [arch=amd64 signed-by=/usr/share/keyrings/repo-keyring.gpg] http://
↪example.org/ubuntu jammy main
```

patch

```
patch: setup_lang.bash
```

Listing 2: setup_lang.bash

```
#!/bin/bash
cat << EOF | sudo chroot squashfs-root
check-language-support
apt install $(check-language-support)
EOF
```

Server Configuration File

Listing 3: iso.yml

```yaml
# ryo-iso config file
###############################

# Base ISO image to rebuild
image: ubuntu/jammy

# Hardware Architecture
arch: amd64

# Base ISO Variant
#   ['desktop', 'server']
# variant: desktop
variant: live-server

# ISO Label
name: RYO Server Installer

# APT subtasks
apt:
  # add-apt-repository
  repository:
    - universe
    - deb http://archive.ubuntu.com/ubuntu/ bionic-updates universe
    - deb [arch=amd64 signed-by=/usr/share/keyrings/repo-keyring.gpg] http://repo.
↪example.org/ubuntu jammy main

  # apt-get install
  install:
    - git
    - python3-pip

# PIP subtasks
pip:
  # pip3 install
  install:
    - doit
    - ansible

patch: ./post.sh
```

Footnotes

2.2 General Configuration

2.2.1 Environment Variables

While the defaults should work well for most applications, it may be useful to to be able to set environmental variables for containerized builds.

RYO_USER_CONFIG_DIR
: Path to directory containing `config.yml`

RYO_ISO_CONF
Path to `iso.yml`

RYO_BUILD_DIR
Build Directory

2.2.2 Program Configuration

Operation of `ryo-iso` is configured by the parameters located in the first config file found.

Config File Search Order
${RYO_USER_CONFIG_DIR}/config.yml
${XDG_CONFIG_HOME}/ryo-iso/config.yml
~/.config/ryo-iso/config.yml

Listing 4: ~/.config/ryo-iso/config.yml

```
distro_image_dir: ~/Downloads
progress: True
debug: True
```

Parameters

Parameter	Default	Description
distro_image_dir	~/Downloads	Download directory for upstream ISO images
progress	True	Display progress indicator
debug	True	Set 'DEBUG' logger level

2.2.3 Project Defaults

Project defaults can be set for all projects via the first config found.

Project Defaults File Search Order
${RYO_USER_CONFIG_DIR}/iso_base.yml
${XDG_CONFIG_HOME}/ryo-iso/iso_base.yml
~/.config/ryo-iso/iso_base.yml

Listing 5: ~/.config/ryo-iso/iso_base.yml

```
pip:
  version: 3
  target: /usr/local/lib/python3.8/dist-packages
  cache: root/.cache/pip

name: 'RYO-ISO'
```

Parameters

Parameter	Default	Description
pip['version']	3	pip version
pip['target']	/usr/local/lib/python3.8/dist-packages	Installation target directory
pip['cache']	root/.cache/pip	Chrooted pip installation dir
name	RYO-ISO	Volume name of the generated ISO

Note

More detailed information about configuration processing can be found in the source.

CHAPTER 3

Changelog

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog and this project adheres to Semantic Versioning.

3.1 [0.1.7]

3.1.1 Changed

- Refactored cleanup of large APT installs
- Improve slugification of https based APT repo names

3.2 0.1.6

3.2.1 Tested

- Built ROS2 + Ubuntu Jammy image on Ubuntu 18.04

3.2.2 Changed

- Curl options updated

3.2.3 Added

- Access to /run/dbus in build chroot NOTE: maybe this is a terrible idea, feedback?

3.2.4 Removed

- APT cache DNS requirement

3.3 0.1.5

3.3.1 Added

- Support for Ubuntu 'jammy'

3.3.2 Changed

- APT keys in `./keys/*.gpg` install target changed for Ubuntu >= 'Jammy' "squashfs-root/etc/apt/trusted.gpg.d/." -> "squashfs-root/usr/share/keyrings/."

3.3.3 Removed

- Deprecated support for Ubuntu 16.04
- Deprecating Python3.6 support

3.4 0.1.4

3.4.1 Fixed

- Locking pydoit to v0.34.2 before deprecating python3.6 support

3.5 0.1.3

3.5.1 Added

- Invalidate sudo credential cache
- `iso.yml` config option for qemu args

3.5.2 Changed

- Optionalize image/casper/filesystem.size update for ubuntu-live-server

3.6 0.1.2

3.6.1 Changed

- Updated requirements in documentation

3.6.2 Fixed

- 'xenial' open(pathlib.Path()) regression
- 'xenial' resolv.conf uptodate check

3.7 0.1.1

3.7.1 Fixed

- Updated documentation for PyPi

3.8 0.1.0

3.8.1 Added

- Project Initialized

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Index

E

environment variable
    RYO_BUILD_DIR, 6
environment variable
    RYO_BUILD_DIR, 9
    RYO_ISO_CONF, 5, 8
    RYO_USER_CONFIG_DIR, 8

R

RYO_BUILD_DIR, 6
RYO_ISO_CONF, 5