A quick guide to kworkflow

What is kw?

This set of scripts have a simple mission: reduces the environment and setup overhead for developing for GNU/Linux. Kw is composed of different scripts unified in a single interface after the installation, kw commands become available in the command line interface.

Install and Uninstall

Package Dependencies:
`apt install libguestfs-tools qemu qemu-kvm python-docutils rsync`
For installing, you just need to run: `./setup -i`. Below you can see all the available options:

Option	*Description*
`--help,-h`	Display this usage message
`--install,-i`	Install kw
`--uninstall,-u`	Uninstall kw
`--completely-remove`	Completely remove everything
`--html`	Build kw's documentation as HTML

Find help

If you need help, you have many options. Via the command line, you can use:
`kw help`
`kw man`
Online:
https://siqueira.tech/doc/kw/
https://github.com/kworkflow/kworkflow

Check codestyle

`codestyle,c`: Apply checkpatch on directory, file, or patch. Example:

```
cd drivers/gpu/drm/amd/display/
kw c amdgpu_dm/amdgpu_dm_irq.h
```

Find maintainers

`maintainers,m` : This command shows the maintainers of a given Kernel module.

Options:

`--authors,-a` : Return the maintainers and the mailing list. "-a" also prints files authors

Example:

```
kw m drivers/gpu/drm/vkms/
kw m -a drivers/gpu/drm/vkms/
```

Find string match

`explore,e` : The explore command is based on git grep. It can search for string matches in either the git repository contents or in the git log messages.

Options:

`STRING [PATH]` : Search for STRING based in PATH (./ by default)

`"STR SRT" [PATH]` : Search for strings

`--log STRING` : Search for STRING on git log

Example:

```
kw e "Atomic check stopped"
kw e "Atomic check stopped" drivers/gpu/drm/
kw e dm_crtc_get_scanoutpos drivers/gpu/drm/
kw e --log "-EINVAL if something gets wrong"
```

Manage .config file

`configm,g` : The 'configm' command manages different versions of the project's '.config' file. It provides the save, load, remove, and list operations of such files.

Options:

`--save NAME [-d 'DESCRIPTION']` : Searches the current directory for a `.config` file to be kept under the management of kw

`--ls` : List config files under kw management

`--get NAME` : Get a config file based named *NAME*

`--rm` : Remove config labeled with *NAME*

Example:

```
cd KERNEL_PATH
kw g --save my_current_config
kw g --ls
kw g --get my_current_config
```

Build

`kw b` or `kw build`

Deploy new Kernel/module

`deploy,d` : If you are in a kernel directory, this command will try to install the current kernel version in your target machine (remote, host, and VM).

Options:

`--remote [REMOTE:PORT]` : Specify the deploy to a remote machine.

`--local` : Deploy in the host machine.

`--vm` : Deploy in the QEMU vm.

`--reboot` : Reboot machine after deploy.

`--modules` : Only deploy modules.

SSH

`ssh,s` :

Options:

`--script,-s [SCRIPT PATH]` : Expects a bash script as a parameter to evaluate in a target machine.

`--command,-c=[COMMAND]` : Expects a command to be executed in a target machine.

Example:

```
kw s
kw s -c="dmesg -wH"
```

kworkflow.config options

`ssh_ip` : Default ssh ip

`ssh_port` : Default ssh port

`arch` : Specify the default architecture used by KW

`virtualizer` : Defines the virtualization tool that should be used by kw. Current, we only support QEMU

`mount_point` : Defines the kw mount point, this directory is used by libguestfs during the mount/umount operation of a VM

`qemu_hw_options` : Sets basic QEMU options

`qemu_net_options` : Defines the network configuration

`qemu_path_image` : Specify the VM image path

`alert` : Default alert options (You should use vs, s, v or n. See README.md for details on this options)

`sound_alert_command` : Command to run for sound completion alert (This command will be executed in the background)

`visual_alert_command` : Command to run for visual completion alert (This command will be executed in the background)

`default_deploy_target` : Sometimes it could be bothersome to pass the same parameter for kw deploy; here, you can set the default target. We define 'vm' as the default, but you can also use 'local' and 'remote REMOTE:PORT'.

`reboot_remote_by_default` :