```
#!/bin/bash
# shellcheck source=kz_common.sh
##############################################################################
# Distribute kz package.
#
# Written in 2009 by Karel Zimmer <info@karelzimmer.nl>, Creative Commons
# Public Domain Dedication <https://creativecommons.org/publicdomain/zero/1.0>.
##############################################################################

set -o errexit
program_path=$(cd "$(dirname "$(realpath "$0")")" && pwd)
source "$program_path"/kz_common.sh


##############################################################################
# Constants
##############################################################################

declare program_name='kz-deploy'
declare program_desc
program_desc=$(gettext 'Distribute kz package')
declare display_name=${program_name/kz-/kz }

declare usage
usage=$(eval_gettext "Usage: \$display_name \$options_usage")
declare help
help="$(eval_gettext "Usage: \$display_name [OPTION...]")

$program_desc.

$(gettext 'Options:')
$options_help"

declare deb_repo=$HOME/kz-deb
declare docs_repo=$HOME/kz-docs
declare scripts_repo=$HOME/kz-scripts
declare uploads_repo=$HOME/kz-uploads
declare uploads_repo_distdir=$uploads_repo/dist


##############################################################################
# Variables
##############################################################################


##############################################################################
# Functions
##############################################################################

function check_input {
    local -i getopt_rc=0
    local parsed=''

    parsed=$(
        getopt  --alternative                       \
                --options       "$options_short"    \
                --longoptions   "$options_long"     \
                --name          "$display_name"     \
                --              "$@"
        ) || getopt_rc=$?
    if [[ $getopt_rc -ne $ok ]]; then
        info "$usage_line"
        exit $error
    fi
```

```
    eval set -- "$parsed"
    kz_common.process_options "$@"

    while true; do
        case $1 in
            --)
                shift
                break
                ;;
            *)
                shift
                ;;
        esac
    done

    if [[ "$*" ]]; then
        printf  "$display_name: $*: %s\n$usage_line\n" \
                "$(gettext 'arguments are not allowed')"
        exit $error
    fi
}


function process_input {
    if groups "$USER" | grep --quiet --regexp='sudo'; then
        sudo true
    else
        printf  '%s\n' "$(gettext 'Already performed by the administrator.')"
        term_script
    fi
    info "* ${bold}Check${normal} ($display_name)"
    check_branch
    info "\n* ${bold}Build${normal} final ($display_name)"
    build_package
    info "\n* ${bold}Deploy${normal} ($display_name)"
    upload_website
    info "\n* ${bold}Install${normal} ($display_name)"
    install_package
}


function check_branch {
    info "$(gettext '* Check that all repos are on branch main...')"
    for repo in  $deb_repo $docs_repo $scripts_repo $uploads_repo; do
        cd "$repo"
        if [[ $(git branch --show-current) != 'main' ]]; then
             warning "$(eval_gettext "Repo \$repo not on branch main.")"
        fi
    done

}


function build_package {
    local -i check_rc=0

    info "$(gettext '* Build package and website (kz build)...')
"
    # Call kz-build.
    "$scripts_repo"/usr/bin/kz build || check_rc=$?
    if [[ $check_rc -ne $ok ]]; then
         error "
$(gettext 'Fix all the messages above and then restart the deploy.')"
        exit $check_rc
```

```
    fi
}


function upload_website {
    local ftp_set='set ssl:verify-certificate no'
    local ftp_from=$uploads_repo_distdir
    local ftp_to=/httpdocs
    local ftp_opts='--reverse --delete --verbose'
    local ftp_exclude='--exclude icaclient_20.04.0.21_amd64.deb'
    local ftp_cmd="mirror $ftp_exclude $ftp_opts $ftp_from $ftp_to; exit"
    local ftp_host=server106.hosting2go.nl
    local ftp_user=kzimmer
    local ftp_login=$HOME/.kz-$ftp_host

    info 'Upload website (lftp)...'
    if ! [[ -f $ftp_login ]]; then
        read -rsp "Wachtwoord voor 'ftp://$ftp_host': "
        info "$REPLY" > "$ftp_login"
        printf '\n'
        chmod 'u=rw,g=,o=' "$ftp_login" |& $logcmd
    fi
    if ! lftp   --user      "$ftp_user"             \
                --password  "$(cat "$ftp_login")"   \
                -e          "$ftp_set; $ftp_cmd"    \
                "$ftp_host"; then
        rm      "$ftp_login"
         error "$(gettext 'Website upload failed.')"
        exit $error
    fi
    sleep 5
}


function install_package {
    # Call kz-getdeb via wget as a regular user would do.
    info "$(gettext 'Install package kz (kz getdeb)...')
"
    if ! wget --output-document=- karelzimmer.nl/kz 2> >($logcmd) | bash; then
         error "$(gettext 'Install package kz failed.')"
        exit $error
    fi
}


function term_script {
    exit $ok
}


################################################################################
# Script
################################################################################

function main {
    kz_common.init_script "$@"
    check_input "$@"
    process_input
    term_script
}

main "$@"
```