

Docker: LAB DI3 image CENTOS sous Centos

Objet	Création d'une image de base CentOS
Niveau requis	débutant, avisé
Débutant, à savoir	
Suivi	:DRAFT:

Création d'une image de base CentOS/Scientific Linux CERN (SLC) sous CentOS/RHEL/SLC

moby/contrib/mkimage-yum.sh

```
#!/usr/bin/env bash
#
# Create a base CentOS Docker image.
#
# This script is useful on systems with yum installed (e.g., building
# a CentOS image on CentOS). See contrib/mkimage-rinse.sh for a way
# to build CentOS images on other systems.

set -e

usage() {
    cat <<E00PTS
$(basename $0) [OPTIONS] <name>
OPTIONS:
    -p "<packages>"  The list of packages to install in the container.
                     The default is blank. Can use multiple times.
    -g "<groups>"    The groups of packages to install in the container.
                     The default is "Core". Can use multiple times.
    -y <yumconf>      The path to the yum config to install packages from. The
                     default is /etc/yum.conf for Centos/RHEL and
                     /etc/dnf/dnf.conf for Fedora
    -t <tag>          Specify Tag information.
                     default is reffered at /etc/{redhat,system}-release
E00PTS
    exit 1
}

# option defaults
yum_config=/etc/yum.conf
if [ -f /etc/dnf/dnf.conf ] && command -v dnf &> /dev/null; then
    yum_config=/etc/dnf/dnf.conf
    alias yum=dnf
fi
# for names with spaces, use double quotes (") as install_groups=('Core'
# 'Compute Node')
install_groups=()
install_packages=()
```

```
version=
while getopts ":y:p:g:t:h" opt; do
    case $opt in
        y)
            yum_config=$OPTARG
            ;;
        h)
            usage
            ;;
        p)
            install_packages+=("\ "$OPTARG\ ")
            ;;
        g)
            install_groups+=("\ "$OPTARG\ ")
            ;;
        t)
            version="$OPTARG"
            ;;
        \?)
            echo "Invalid option: -$OPTARG"
            usage
            ;;
    esac
done
shift $((OPTIND - 1))
name=$1

if [[ -z $name ]]; then
    usage
fi

# default to Core group if not specified otherwise
if [ ${#install_groups[*]} -eq 0 ]; then
    install_groups=('Core')
fi

target=$(mktemp -d --tmpdir $(basename $0).XXXXXX)

set -x

mkdir -m 755 "$target"/dev
mknod -m 600 "$target"/dev/console c 5 1
mknod -m 600 "$target"/dev/initctl p
mknod -m 666 "$target"/dev/full c 1 7
mknod -m 666 "$target"/dev/null c 1 3
mknod -m 666 "$target"/dev/ptmx c 5 2
mknod -m 666 "$target"/dev/random c 1 8
mknod -m 666 "$target"/dev/tty c 5 0
mknod -m 666 "$target"/dev/tty0 c 4 0
mknod -m 666 "$target"/dev/urandom c 1 9
```

```
mknod -m 666 "$target"/dev/zero c 1 5

# amazon linux yum will fail without vars set
if [ -d /etc/yum/vars ]; then
    mkdir -p -m 755 "$target"/etc/yum
    cp -a /etc/yum/vars "$target"/etc/yum/
fi

if [[ -n "$install_groups" ]];
then
    yum -c "$yum_config" --installroot="$target" --releasever=/ --
setopt=tsflags=nodocs \
        --setopt=group_package_types=mandatory -y groupinstall
"${install_groups[*]}"
fi

if [[ -n "$install_packages" ]];
then
    yum -c "$yum_config" --installroot="$target" --releasever=/ --
setopt=tsflags=nodocs \
        --setopt=group_package_types=mandatory -y install
"${install_packages[*]}"
fi

yum -c "$yum_config" --installroot="$target" -y clean all

cat > "$target"/etc/sysconfig/network <<EOF
NETWORKING=yes
HOSTNAME=localhost.localdomain
EOF

# effectively: febootstrap-minimize --keep-zoneinfo --keep-rpmdb --keep-
services "$target".
# locales
rm -rf
"$target"/usr/{lib,share}/locale,{lib,lib64}/gconv,bin/localedef,sbin/build
-locale-archive}
# docs and man pages
rm -rf "$target"/usr/share/{man,doc,info,gnome/help}
# cracklib
rm -rf "$target"/usr/share/cracklib
# i18n
rm -rf "$target"/usr/share/i18n
# yum cache
rm -rf "$target"/var/cache/yum
mkdir -p --mode=0755 "$target"/var/cache/yum
# sln
rm -rf "$target"/sbin/sln
# ldconfig
rm -rf "$target"/etc/ld.so.cache "$target"/var/cache/ldconfig
mkdir -p --mode=0755 "$target"/var/cache/ldconfig
```

```
if [ -z "$version" ]; then
    for file in "$target"/etc/{redhat,system}-release
    do
        if [ -r "$file" ]; then
            version="$(sed 's/^[^0-9\]*\([0-9.\]\+\).*$/\1/' "$file")"
            break
        fi
    done
fi

if [ -z "$version" ]; then
    echo >&2 "warning: cannot autodetect OS version, using '$name' as tag"
    version=$name
fi

tar --numeric-owner -c -C "$target" . | docker import - $name:$version

docker run -i -t --rm $name:$version /bin/bash -c 'echo success'

rm -rf "$target"
```

febootstrap-minimize

La commande unix febootstrap-minimise minimise un système de fichiers créé par febootstrap, ce qui signifie que les fichiers inutiles sont supprimés de l'image.

Si aucune option n'est donnée, la valeur par défaut est de minimiser autant que possible l'image, ce qui signifie, par exemple, que les paramètres régionaux sont supprimés de sorte que l'image ne sera utilisable qu'en anglais américain, sans documentation ni pages de manuel. L'image ne fonctionnera que dans un fuseau horaire UTC. Pour conserver ces paramètres on peut spécifier les options telles que:

```
febootstrap-minimize --keep-zoneinfo --keep-rpmdb --keep-services "$target"
```

- **-keep-locales:** Conserve le support des paramètres régionaux.
- **-keep-zoneinfo:** Conserve tous les fuseaux horaires
- **-keep-rpmdb:** Conserve les bases de données des packages RPM et YUM.
- **-keep-services:** Conserve le fichier /etc/services. Si on le supprime, ce fichier est remplacé par un fichier très minime qui répertorie uniquement les services les plus courants. Pour les services moins courants, on devra s'y référer par numéro de port plutôt que par nom.

Dans l'exemple intègre les commandes rm pour minimiser l'image construite plutôt que cette commande.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=labs:docker-lab-di3-centos-from-centos>

Last update: **2025/02/19 10:59**

