

# Disk Monitoring Skript

The script `docker_image_backup.sh` creates a local copy of the docker images and save them under `/data01/save/`. With the script you have the possibility to save (docker save) or load (docker load) docker images.

## How to Use

```
[root@v81elt dc-config]# ./docker_image_backup.sh -h
Usage: ./docker_image_backup.sh [OPTIONS] [IMAGE:TAG]
Example: ./docker_image_backup.sh -s
docker.pnet.ch/toolchain/artifactory:5.4.6-0
OPTIONS:
  -s | --save          Save image (backup) [IMAGE:TAG] or [all]
  -l | --load          Load image (recover) [IMAGE:TAG]
  -d | --delete        Delete older images (backuper) than 90days
  -h | --help          Help
[root@v81elt dc-config]# docker images | grep artifactory
docker.pnet.ch/tool/artifactory          5.8.3-ha
5ccd33d863cc          3 weeks ago          657 MB
[root@v81elt dc-config]# ./docker_image_backup.sh -s
docker.pnet.ch/tool/artifactory:5.8.3-ha
Create backup of docker.pnet.ch/tool/artifactory:5.8.3-ha to /data01/save/
...
Backup created successfully.
[root@v81elt dc-config]# ls -la /data01/save/ | grep artifactory
-rw----- . 1 root docker 669214720 Mar 19 08:50
docker.pnet.ch.tool.artifactory.5.8.3-ha.tar
```

## Skript Sourcecode

Filename: **docker\_image\_backup.sh**

```
#!/bin/bash
#
# Usage:          ./docker_image_backup.sh [OPTIONS] [IMAGE:TAG]
#
# Documentation:
https://wikit.post.ch/display/DEVPLAT/Local+Docker+Image+Backup
#
# Author:         Jan Schneider, jan.schneider@post.ch, I353
#
```

```
# -----  
-----  
# Edition history:  
#  
# Version      Date        Who      What  
# 0.0.1        13.03.18   SchJ     initial version  
# 0.0.2        26.03.18   SchJ     remove function  
  
dockerbin=`which docker`  
if [ -z "$dockerbin" ]; then echo "No docker binary found"; exit 1; fi  
  
savedir=/data01/save/  
[[ -d $savedir ]] || mkdir -p $savedir  
  
function usage {  
# View the usage message  
  
echo -e "\e[1;35mUsage: ./docker_image_backup.sh [OPTIONS] [IMAGE:TAG]\e[0m"  
cat <<EOF  
Example: ./docker_image_backup.sh -s  
docker.pnet.ch/toolchain/artifactory:5.4.6-0  
  
OPTIONS:  
-s | --save          Save image (backup) [IMAGE:TAG] or [all]  
-l | --load          Load image (recover) [IMAGE:TAG]  
-d | --delete        Delete older images (backupid) than 90days  
-h | --help          Help  
EOF  
}  
  
function delete {  
    find $savedir -type f -mtime +90 -delete  
}  
  
function save {  
    tarfile=`echo $1 | tr -s "/" "."`.tar  
    echo "Create backup of $1 to $savedir ..."  
    $dockerbin save -o $savedir/$tarfile $1  
  
    if [ $? -eq 0 ]; then  
        echo "Backup created successfully."  
    else  
        echo "Backup creation failed!"  
        exit 1  
    fi  
}  
  
function save_all {
```

```
    for image in `$dockerbin images | tr -s " " | cut -d" " -f-2 | grep -v
"REPOSITORY TAG" | tr " " ":"`; do
        save $image
    done
}

function load {
    tarfile=`echo $1 | tr -s "/" "."`.tar
    if [ ! -f $savedir/$tarfile ]; then
        echo "Backup tar not found!"
        exit 1
    fi
    echo "Loading backup tar of $1 ..."
    $dockerbin load < $savedir/$tarfile
    if [ $? -eq 0 ]; then
        echo "Backup successfully loaded."
    else
        echo "Backup loading failed!"
        exit 1
    fi
}

while true; do
    case "$1" in
        -h|--help)
            usage
            shift
            ;;
        -d|--delete)
            delete
            shift
            ;;
        -s|--save)
            shift
            if [[ "$1" = "all" ]]; then
                save_all
                shift
            else
                while [[ "$1" != "-"* && "$1" != "" ]]; do
                    save $1
                    shift
                done
            fi
            ;;
        -l|--load)
            shift
            while [[ "$1" != "-"* && "$1" != "" ]]; do
                load $1
                shift
            done
        ;;
    esac
done
```

```
done
;;
*) exit 0 ;;
esac
done
```

Last update: **2019/02/11 14:11**