

Borg Backup for Container Config



Install latest release of <https://www.borgbackup.org/> from <https://github.com/borgbackup/borg/releases>

- custom backup scripts to backup container configs from `/etc/svc/config/${APPNAME}`

1. Initialization of backups

- create a new borg repository locally in `/dev/shm/${APPNAME}`
- Create a new github repository under `pknw1-config/borg-${APPNAME}`
- Push the base files as the initial commit

```
export APPNAME=bookstack
export CONF_FOLDER=/etc/svc/config/${APPNAME}
borg repo-create --encryption repokey-aes-ocb -r /dev/shm/${APPNAME}
gh repo create pknw1-config/borg-${APPNAME} --private
cd /dev/shm/${APPNAME}
git init && git add . && git commit -m "init" && git branch -M main && git remote add origin
git@github.com:pknw1-config/borg-${APPNAME}.git && git push -u origin main
```

- create an initial backup

```
borg -p -r /dev/shm/${APPNAME} create ${APPNAME} -e '*MediaCover*' -e '*Logs*' -e '*logs*' -e
'*Backups*' -e '*cache*' -e 'metadata' --stats "init" ${CONF_FOLDER}
```

2. Regular Backups

borg-docker-auto.sh

```
#!/bin/bash
BRANCH=$(date +%s)
COMMIT=$BRANCH
export BORG_PASSPHRASE=****secret****

if [[ $# -eq 0 ]]
then
    echo "no application specified - trying to guess"

    GUESS=$(pwd | awk -F/ '{print $NF}')
    if [[ -d /etc/svc/config/${GUESS} ]] || [[ -d /etc/pkwn1/config/${GUESS} ]]
    then
        echo " ${GUESS}"
        read -N 1 -p "Guessed App ${GUESS} found : Proceed? [Y/N] : " PROCEED
        PROCEEDU=$(echo ${PROCEED} | tr '[:lower:]' '[:upper:]')
        if [[ ${PROCEEDU} == "Y" ]]
        then
            echo "Y"
            APPNAME=${GUESS}
        else
            echo "N"
            exit 0
        fi
    else
        echo "could not guess"
    fi
else
    echo "appname ${1} supplied"
    APPNAME=$1
fi

REMOTE_REPO=$(git ls-remote git@github.com:pkwn1-config/borg-${APPNAME}.git)
if ! [[ -z ${REMOTE_REPO} ]]
then
    echo "repo found"
```

```

else
    echo "no repo"
    exit 128
fi

[[ -d /etc/svc/config/${APPNAME} ]] && read -p "Using config folder
/etc/svc/config/${APPNAME} - or enter alternative locatioon: " CONF_FOLDER_OVERRIDE

if [[ -z ${CONF_FOLDER_OVERRIDE} ]]
then
    echo "using default"
    CONF_FOLDER=/etc/svc/config/${APPNAME}
else
    while ! [[ -d ${CONF_FOLDER_OVERRIDE} ]]
    do
        read -p "    Error ${CONF_FOLDER_OVERRIDE} folder not found - enter a new location
: " CONF_FOLDER_OVERRIDE
        if [[ -z ${CONF_FOLDER_OVERRIDE} ]]
        then
            echo "no location"
            exit 127
        fi
    done
    echo "using ${CONF_FOLDER_OVERRIDE}"
    CONF_FOLDER=${CONF_FOLDER_OVERRIDE}
fi

echo $CONF_FOLDER

MATCHED_CONTAINERS=$(docker ps | grep $APPNAME | awk '{print $2}' )
MATCHED_CONTAINERS_COUNT=$(docker ps | grep $APPNAME | awk '{print $2}'|wc -l )

case $MATCHED_CONTAINERS_COUNT in
    "0") echo "Error zero"
        exit 128
        ;;
    "1") VALID_LIST=[]
        VALID_LIST+=(${MATCHED_CONTAINERS})
        ;;

```

```

*) echo $MATCHED_CONTAINERS_COUNT
   VALID_LIST=[]
   FIRST_CONTAINER=$(docker ps | grep $APPNAME | awk '{print $2}'|head -n1)
   PROJECT=$(docker inspect ${FIRST_CONTAINER}| grep '"com.docker.compose.project"' |
awk -F\" '{print $4}')

   for TEST_CONTAINER in $(docker ps | grep $APPNAME | awk '{print $2}')
   do
       PROJECT_TEST=$(docker inspect ${TEST_CONTAINER}| grep
'"com.docker.compose.project"' | awk -F\" '{print $4}')
       if [[ ${PROJECT_TEST} == ${PROJECT} ]]
       then
           #echo "add"
           VALID_LIST+=(${TEST_CONTAINER})
       fi
   done

;;
esac

for CONTAINER in "${VALID_LIST[@]}" # or you can use "${stringPhrases[*]}"
do
    echo "Stopping $CONTAINER";
    docker stop ${CONTAINER}
done

[[ -d /dev/shm/${APPNAME} ]] || git clone git@github.com:pknw1-config/borg-${APPNAME}.git
/dev/shm/${APPNAME}

cd /dev/shm/${APPNAME} && git checkout -b ${BRANCH}
borg -p -r /dev/shm/${APPNAME} create ${APPNAME} -e '*MediaCover*' -e '*Logs*' -e '*logs*'
-e '*Backups*' -e '*cache*' -e 'metadata' --stats $COMMIT ${CONF_FOLDER}
git add . && git commit -m ${COMMIT} && git push -u origin ${BRANCH}

for CONTAINER in "${VALID_LIST[@]}" # or you can use "${stringPhrases[*]}"
do
    echo "Starting $CONTAINER";
    docker start ${CONTAINER}
done

```

Using the backup script `borg-docker-auto.sh ${APPNAME}` will

- find all running containers under the app project and `docker stop` each container
- if `/dev/shm/${APPNAME}` doesnt exist, it will clone the remote repo `pknw1-config/borg-${APPNAME}` is cloned locally
- a nw git branch is created
- borg backup from `/etc/svc/config/${APPNAME}` is done to the folder
- all files are `git add`, `git commit` and `git push` to the new branch
- each of the stopped containers are re-restarted with `docker start`

Revision #1

Created 2025-07-17 00:24:36 CEST by pkswansea

Updated 2025-07-17 00:37:44 CEST by pkswansea