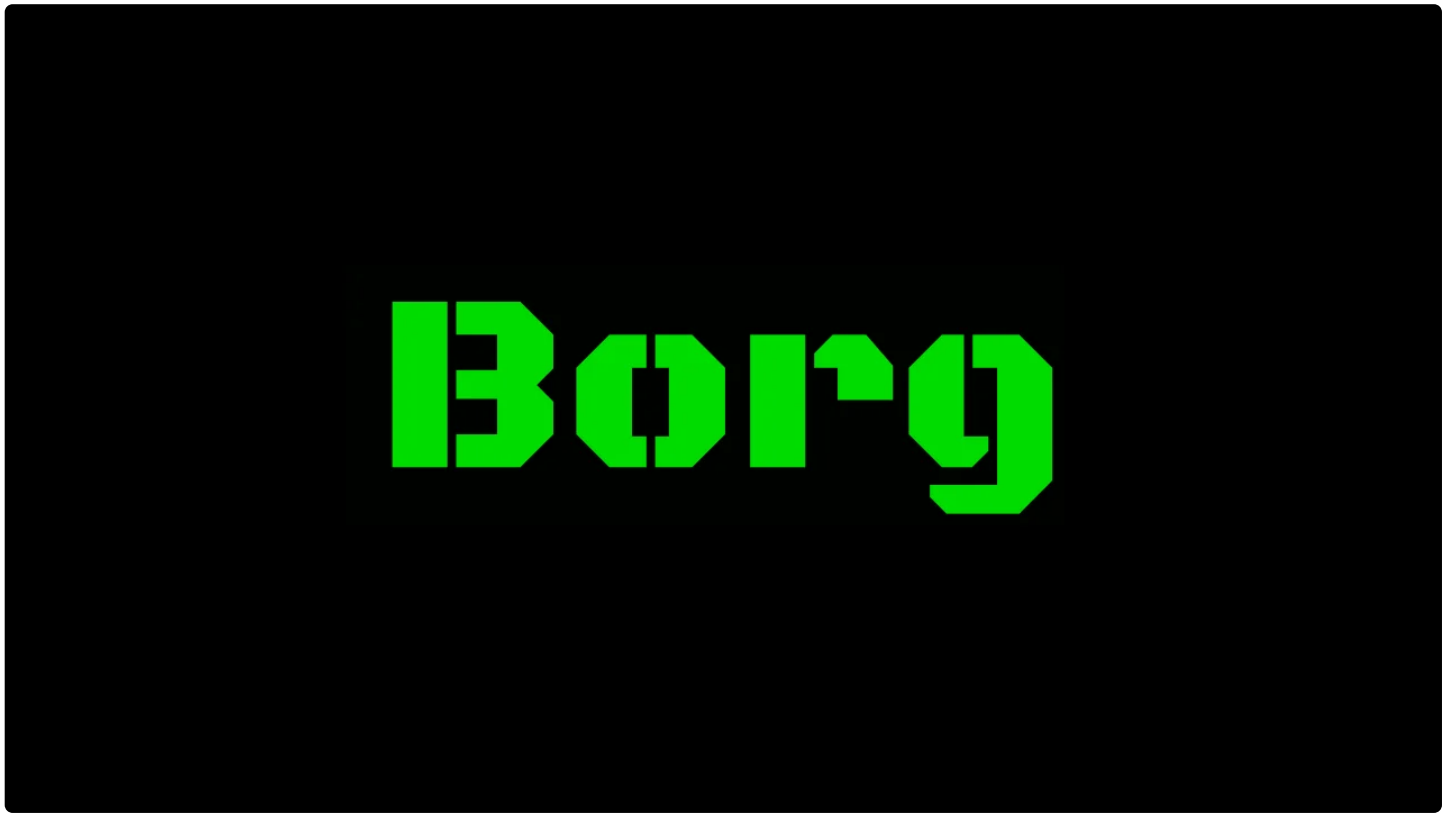
Sysadmin

Set up automated server backups with Borg

The Borg deduplicating backup program can automate daily, weekly, and monthly backups with a single script saving space and keeping data safe from mistakes.



Author

Isaac Bythewood

June 25, 2022

Borg is a backup program written in Python that has excellent deduplication functionality. For my blog server backups you can see just how much smaller the deduplicated repository size is.

```
------------------------------------------------------------------------------
                           Original size      Compressed size    Dec
All archives:                780.76 MB            588.23 MB          14

                       Unique chunks         Total chunks
Chunk index:                    2780                24029
------------------------------------------------------------------------------
```

If you run your websites on small servers like I do then this comes in handy to save some money. I currently have two layers of backups on my servers:

1. Linode provides their "Linode Backup" system. This is a complete backup of the entire disk your server is running on for a reasonable price.
2. Borg backup for daily automated backups that allow for quick "oops" restores.

These backups serve two different purposes. I can do an emergency full system restore using Linode, and a quick Borg restore if I happen to do something stupid, like accidently delete a blog post.

I currently use a slightly modified script in my `/etc/periodic/daily` folder based on [Borg's quick start automation suggestion](#).

```
#!/bin/sh

# Setting this, so the repo does not need to be given on the commandline
export BORG_REPO=/srv/backup

# some helpers and error handling:
info() { printf "\n%s %s\n\n" "$( date )" "$*" >&2; }
trap 'echo $( date ) Backup interrupted >&2; exit 2' INT TERM

info "Starting backup"
```

```
11
12 # Backup the most important directories into an archive named after
13 # the machine this script is currently running on:
14
15 borg create                          \
16     --verbose                         \
17     --filter AME                      \
18     --list                            \
19     --stats                           \
20     --show-rc                         \
21     --compression lz4                 \
22     --exclude-caches                  \
23                                       \
24     ::'{now}'                         \
25     /srv/git                          \
26     /srv/docker                       \
27     /srv/data                         \
28     /etc/caddy                        \
29
30 backup_exit=$?
31
32 info "Pruning repository"
33
34 # Use the `prune` subcommand to maintain 7 daily, 4 weekly and 6 monthly
35 # archives of THIS machine.
36
37 borg prune                          \
38     --list                            \
39     --show-rc                         \
40     --keep-daily    7                 \
41     --keep-weekly   4                 \
42     --keep-monthly  6                 \
43
44 prune_exit=$?
45
46 # actually free repo disk space by compacting segments
47
48 info "Compacting repository"
49
50 borg compact
```

```

compact_exit=$?

# use highest exit code as global exit code
global_exit=$(( backup_exit > prune_exit ? backup_exit : prune_exit ))
global_exit=$(( compact_exit > global_exit ? compact_exit : global_exit

if [ ${global_exit} -eq 0 ]; then
    info "Backup, Prune, and Compact finished successfully"
elif [ ${global_exit} -eq 1 ]; then
    info "Backup, Prune, and/or Compact finished with warnings"
else
    info "Backup, Prune, and/or Compact finished with errors"
fi

exit ${global_exit}
```

Borg's documentation can explain this better than I can but the gist is:

- I run daily backups on all my unique server files in `/srv` and `/etc`. I don't backup anything that is a system default.
- Then prune my backups to only keep 7 daily, 4 weekly, and 6 monthly backups at any given time.
- Then we compact the repository to save space and exit.

A better option for storing Borg backups would be to set up a Borg repo on another server or a platform like BorgBase. BorgBase is nice since they will notify you by email if a backup doesn't happen or fails however, as I said before, I only use Borg for "oops" backups so if I were to miss a couple I wouldn't stress out about it.

As an extra to this post, I have on my servers a `server-health-check.sh` script that allows me to quickly check things like auto-updates, Borg backups, free memory, and used disk space! I run it every so often just for peace of mind.

```
#!/bin/sh

echo -e "\napk upgrades ----------------------------------------------
tail /var/log/apk-autoupgrade.log

echo -e "\nborg backups ----------------------------------------------
borg list /srv/backup

echo -e "\nfree memory  ----------------------------------------------
free -h | head -n2

echo -e "\nfree space   ----------------------------------------------
df -h | head -n1 && df -h | grep "/dev/sda" | head -n1
```

The output of this script is pretty well formatted without much effort and looks a little something like this:

```
apk upgrades ------------------------------------------------------------
[Sat Jun 25 02:00:00 UTC 2022] fetch http://dl-cdn.alpinelinux.org/alpi
[Sat Jun 25 02:00:00 UTC 2022] fetch http://dl-cdn.alpinelinux.org/alpi
[Sat Jun 25 02:00:00 UTC 2022] OK: 512 MiB in 253 packages

borg backups ------------------------------------------------------------
2022-06-19T02:00:00                  Sun, 2022-06-19 02:00:01 [1c4a6d43
2022-06-20T02:00:01                  Mon, 2022-06-20 02:00:01 [cf62b244
2022-06-21T02:00:07                  Tue, 2022-06-21 02:00:08 [45001602
2022-06-22T02:00:01                  Wed, 2022-06-22 02:00:01 [85aa754d
2022-06-23T02:00:01                  Thu, 2022-06-23 02:00:01 [6014ed6f
2022-06-24T02:00:01                  Fri, 2022-06-24 02:00:01 [5676379d
2022-06-25T02:00:00                  Sat, 2022-06-25 02:00:01 [8a5c05fc

free memory  ------------------------------------------------------------
              total        used        free      shared  buff/cache   av
Mem:         983.8M      626.2M       97.1M      516.0K      260.5M

free space   ------------------------------------------------------------
Filesystem                Size      Used Available Use% Mounted on
/dev/sda                 24.1G      7.5G     15.4G  33% /
```

If you're interested in Borg you can read more about it on Borg's website, they have extensive and well written documentation.
