

GitLab - Restore Backup

GitLab allows restoring the backup copy of your repository. In this chapter, we will discuss about how to restore the backup copy in the GitLab –

Step 1 – First, login to your GitLab server using SSH (Secure Shell).

Step 2 – Before restoring the backup copy, first make sure backup copy is in the */var/opt/gitlab/backups* directory.

Step 3 – You can check the backup copy by using the *ls -l* command which is described in the Create Backup job chapter.

Step 4 – Now, stop the processes which are related to the database by using the below commands –

```
sudo gitlab-ctl stop unicorn  
  
sudo gitlab-ctl stop sidekiq
```

```
root@buds_gitlab:~# sudo gitlab-ctl stop unicorn  
ok: down: unicorn: 9013s, normally up  
root@buds_gitlab:~# sudo gitlab-ctl stop sidekiq  
ok: down: sidekiq: 9008s, normally up
```

The above commands can also be used to free up some memory temporarily by shutting down them.

Step 5 – You can verify status of the GitLab services by using the below command –

```
sudo gitlab-ctl status
```

Step 6 – Now, restore the backup by using the timestamp of the backup copy –

```
sudo gitlab-rake gitlab:backup:restore BACKUP = 1521884424_2018_03_24_10.5.3
```

```

root@buds_gitlab:~# sudo gitlab-rake gitlab:backup:restore BACKUP=1521884424_20
18_03_24_10.5.3
Unpacking backup ... done
Restoring repositories ...
* root/first-gitlab-prjt ... [DONE]
* root/project_demo ... [DONE]
* root/ny-awesome-project ... [DONE]
Put GitLab hooks in repositories dirs [DONE]
done
Restoring builds ...
done
Restoring artifacts ...
done
Restoring pages ...
done
Restoring lfs objects ...
done
This will rebuild an authorized_keys file.
You will lose any data stored in authorized_keys file.
Do you want to continue (yes/no)? yes

Deleting tmp directories ... done
done
done
done
done
done

```

Step 7 – Restart the GitLab components by using the below command –

```
sudo gitlab-ctl restart
```

```

root@buds_gitlab:~# sudo gitlab-ctl restart
ok: run: gitaly: (pid 23900) 1s
ok: run: gitlab-monitor: (pid 23912) 0s
ok: run: gitlab-workhorse: (pid 23924) 1s
ok: run: logrotate: (pid 23935) 0s
ok: run: nginx: (pid 23941) 1s
ok: run: node-exporter: (pid 23947) 0s
ok: run: postgres-exporter: (pid 24026) 0s
ok: run: postgresql: (pid 24034) 1s
ok: run: prometheus: (pid 24042) 0s
ok: run: redis: (pid 24051) 1s
ok: run: redis-exporter: (pid 24055) 0s
ok: run: sidekiq: (pid 24060) 0s
ok: run: unicorn: (pid 24067) 1s

```

Step 8 – Now check the GitLab by sanitizing the database as shown below –

```
sudo gitlab-rake gitlab:check SANITIZE = true
```

```

root@buds_gitlab:~# sudo gitlab-rake gitlab:check SANITIZE=true
Checking GitLab Shell ...

GitLab Shell version >= 6.0.3 ? ... OK (6.0.3)
Repo base directory exists?
default... yes
Repo storage directories are symlinks?
default... no
Repo paths owned by git:root, or git:git?
default... yes
Repo paths access is drwxrws---?
default... yes
hooks directories in repos are links: ...
1/1 ... repository is empty
1/2 ... repository is empty
1/3 ... repository is empty
Running /opt/gitlab/embedded/service/gitlab-shell/bin/check
Check GitLab API access: OK
Redis available via internal API: OK

Access to /var/opt/gitlab/.ssh/authorized_keys: OK
gitlab-shell self-check successful

Checking GitLab Shell ... Finished

Checking Sidekiq ...

Running? ... yes
Number of Sidekiq processes ... 1

Checking Sidekiq ... Finished

Reply by email is disabled in config/gitlab.yml
Checking LDAP ...

LDAP is disabled in config/gitlab.yml

Checking LDAP ... Finished

Checking GitLab ...

Git configured correctly? ... yes
Database config exists? ... yes
All migrations up? ... yes
Database contains orphaned GroupMembers? ... no
GitLab config exists? ... yes
GitLab config up to date? ... yes
Log directory writable? ... yes
Tmp directory writable? ... yes
Uploads directory exists? ... yes
Uploads directory has correct permissions? ... yes
Uploads directory tmp has correct permissions? ... skipped (no tmp uploads
yet)
Init script exists? ... skipped (omnibus-gitlab has no init script)
Init script up-to-date? ... skipped (omnibus-gitlab has no init script)
Projects have namespace: ...
1/1 ... yes
1/2 ... yes
1/3 ... yes
Redis version >= 2.8.0? ... yes
Ruby version >= 2.3.5 ? ... yes (2.3.6)
Git version >= 2.9.5 ? ... yes (2.14.3)
Git user has default SSH configuration? ... yes
Active users: ... 1

Checking GitLab ... Finished

```

The `SANITIZE = true` flag removes all email addresses because they are confidential, removes the CI variables and access tokens as they can be used in the production instance.