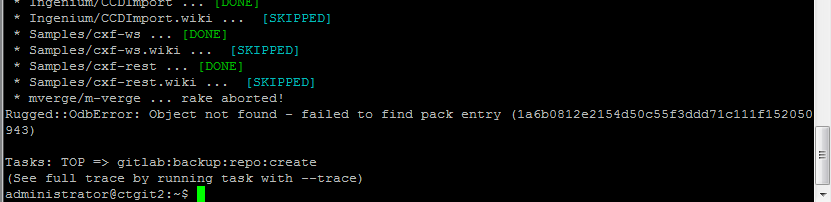
For GitLab Omnibus version so we will use below mentioned command to create a backup of GitLab repositories including Database

**sudo gitlab-rake gitlab:backup:create**

Note: The server requires minimum 2 GB of RAM to create a backup using the above mentioned command

During the backup process gitlab will take individual repositions backup and will skip the wiki page of each repo’s as they doesn’t exists



As currently there are some repositories which are corrupted & giving “**internal server error 500**”, hence we are not able to perform the backup, above is the screen capture of error found during the backup process.

We need to upgrade the current version of Gitlab server to 1.8 and above which will provides us an additional maintenance feature of cleaning repositories, alternatively we can delete unwanted repositions post the backup has completed successfully.

# Upgrade

To upgrade the Gitlab CE edition perform the below setups.

1. Download the Latest Gitlab package from the website [**https://packages.gitlab.com/gitlab/gitlab-ce**](https://packages.gitlab.com/gitlab/gitlab-ce) , if you are using .rpm package refer the below setups to convert rpm package to deb format else skip to setup 4
2. Run the below command to get alien package

**sudo apt-get install alien dpkg-dev debhelper build-essential**

1. **sudo alien packagename.rpm**
2. Run the following command to stop the Gitlab service

**sudo gitlab-ctl stop unicorn**

**sudo gitlab-ctl stop sidekiq**

**sudo gitlab-ctl stop nginx**

1. To install the package, you’ll use the dpkg utility, which is the internal package management tool behind debian and Ubuntu

**sudo dpkg -i packagename.deb**

1. Post successful installation start the following service

**sudo gitlab-ctl start unicorn**

**sudo gitlab-ctl start sidekiq**

**sudo gitlab-ctl start nginx**

1. Reconfigure the Gitlab and verify if the application is running fine

**sudo gitlab-ctl reconfigure**

Post successful upgrade of Gitlab server kindly re-run the backup command and ensure that the backup is completed successfully.

# Backup

**sudo gitlab-rake gitlab:backup:create**

The backup will be stored at path “/var/opt/gitlab/backups”, if you want to store your GitLab backups in a different directory, modify the backup destination path into the file “/etc/gitlab/gitlab.rb” and run **sudo gitlab-ctl reconfigure** command

Once the backup is successfully completed, you can schedule the same using cron job

To schedule the backup using cron job run

1. **sudo su –**
2. **crontab –e**
3. Add the following line to schedule the backup for everyday at 2 AM

**0 2 \* \* \* /opt/gitlab/bin/gitlab-rake gitlab:backup:create CRON=1**

# Restoration

**Note: The below command will overwrite the contents of your GitLab database; kindly ensure that you are using an alternate server (Clone/backup) to restore the Gitlab DB and repositories**

1. First make sure your backup tar file is in the backup directory described in the gitlab.rb configuration.

/var/opt/gitlab/backups

1. Stop the processes that are connected to the database, leave the rest of GitLab running

**sudo gitlab-ctl stop unicorn**

**sudo gitlab-ctl stop sidekiq**

**sudo gitlab-ctl status**

1. Restore the backup, specifying the timestamp of the backup you wish to restore

**sudo gitlab-rake gitlab:backup:restore BACKUP=1393513186**

1. Once you have successfully restored the Gitlab DB and repositories, you can connect the backup/clone server and restore the required branches manually to the production server.