**PHP Build Deployment Steps**

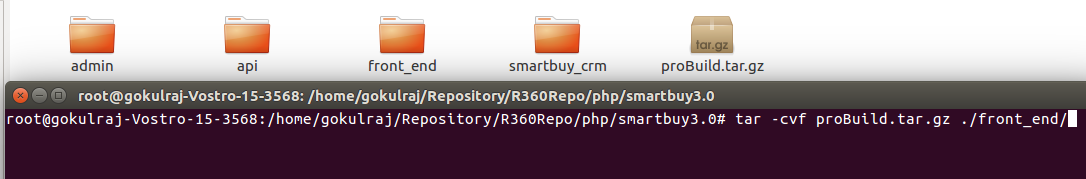
1. Initially, take git pull update from the master branch of the repository and make a note of modified files and directories.

**git pull origin master**

1. Using **tar** command, generate a **tar.gz** file(Eg. Prodbuild.tar.gz) having complete pack of files from the repository.

cd ../Repository/R360Repo/php/smartbuy3.0

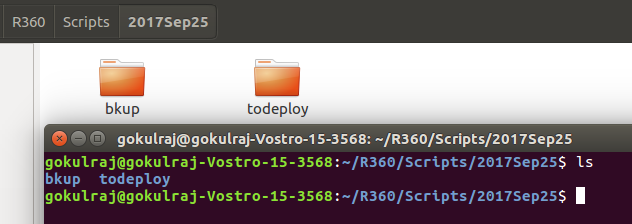
**tar -cvf prodBuild.tar.gz ./front\_end/**

****

1. Update this new Build to **SFTP FileShare.**

**Sftp** [**R360FileShare@52.66.95.143**](mailto:R360FileShare@52.66.95.143)

1. Once you logged into the prod server, In your home directoty, Create a directory with current day’s date and two sub directories namely **backup** and **toBeDeployed**



1. Transfer the updated build from **SFTPFileShare** to **toBeDeployed** folder in the production server.
2. Extract the transferred build file( Prodbuild.tar.gz) in the same folder.

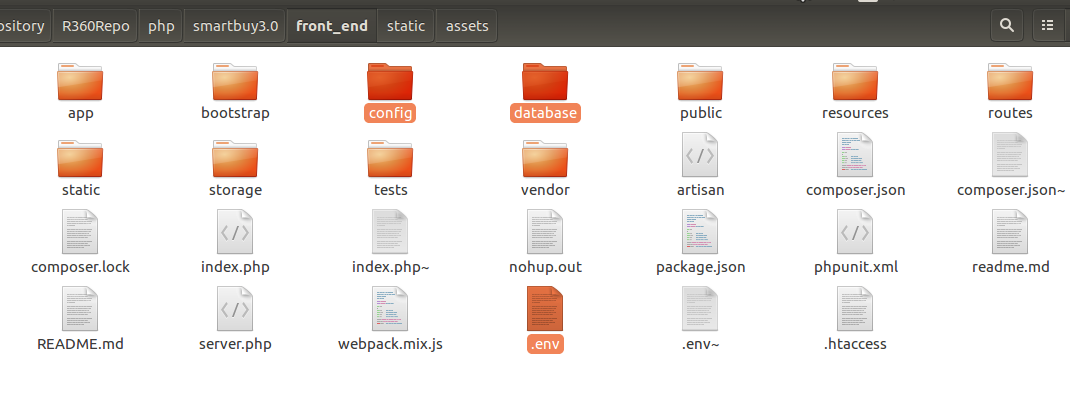
**tar -xzvf prodBuild.tar.gz**

1. Before Deploying, Take a complete Backup of currently running Build in backup folder.
2. Once the Complete Backup is taken, proceed deploying the new build as per below Steps,
3. Below Files/Folders **MUST** **NOT** be changed in Production Server at any Cost

**1. config** folder

**2. database** folder

**3. .env file**



1. Frequent Code update usually happens on the files in below folders

**1. app**

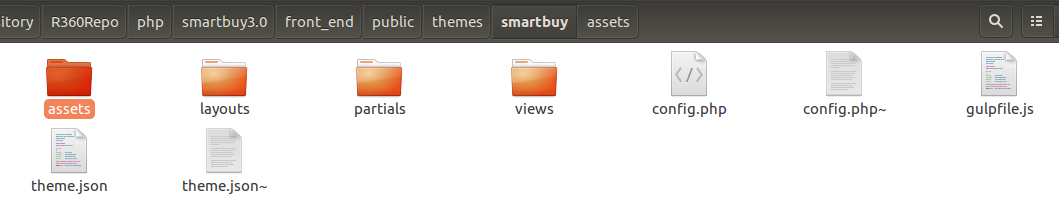
**2. public**

**3. routes**

So generally it is always necessary to check and update these modified files or folders

1. It is **NOT required** to move static data like **/smartbuy/assets/\*** at all times if there is no update from its repo.

**\*\*\*** Assets directories hold static conetents only,hence it is not required to update the same upon every build update.



1. Once the Deployment is done, **Optimize** php and **Clear all php caches** in the Server using below commands,

[root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front\_end#](mailto:root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front_end) **php artisan optimize**   
 *Generating optimized class loader   
 The compiled services file has been removed.*   
 [root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front\_end#](mailto:root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front_end) **php artisan config:cache**   
 *Configuration cache cleared!   
 Configuration cached successfully!*   
 [root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front\_end#](mailto:root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front_end) **php artisan route:cache**   
 *Route cache cleared!   
 Routes cached successfully!*

[*root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front\_end#*](mailto:root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front_end)***php artisan cache:clear*** *Cache cleared successfully.*   
 [root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front\_end#](mailto:root@ip-10-60-2-156:/data/code/webapp/smartbuy3.0/front_end) **php artisan view:clear**   
 View c*ache cleared successfully.*

***php artisan cache:clear***

This does two things.

1. Cache::flush() is called to empty the cache.
2. The app/storage/meta/services.json file is erased. This file is created as Laravel tries to optimize the loading of the service providers your application uses.

Please note that since this command deletes a file from the local file system, if you have multiple servers running your application you must execute it on each server.

**\*\*\*\*cache clear is must to be carried out for every changes in config files.**