

Migrating A LAMP Site To A New Host

A quick reference on how to launch an application on a new hosting provider. For migrating between servers within a hosting provider, reference their documentation or bottom of page 2.

NOTE: Variable values are surrounded in brackets. If your database is called “johnnysdonuts”, [database] would be “johnnysdonuts”. This process uses a local MySQL installation.

1. Make sure your repository is up to date. If not using a repository, skip to step 3.
 - a. On your current server, run “git status” to see if any hot-fixes have not been committed.
 - i. If any files are modified, run:
 1. `$ git add -A`
 2. `$ git commit -m “committing hot fixes”`
 3. `$ git push origin [branch]`
2. Download your site files. To start, log into your new server and open your site root:
 - a. Connect to your instance. **Download** our “Connecting to an AWS Instance” whitepaper on our website or by contacting sales@arcanestrategies.com
 - b. `$ cd [path to webroot]`
 - i. ie. `$ cd /var/www/html/[domain]`
 - c. `$ git init .`
 - d. `$ git clone [your repository]`
 - i. You will be asked to log into your account if using HTTPS. If using SSH, follow instructions on your repository’s website.
 - e. `$ touch .gitignore`
 - i. Create a gitignore file to prevent overwriting config files before you create them. To learn more about adding them now and in the future, see <https://git-scm.com/docs/gitignore>
 - ii. Now create or edit any files that might otherwise not be contained in your repository such as config files or .htaccess files. “Touch” creates new files, vi/vim/nano edits them.
3. (Optional) If no Repository, download via FTP. We assume you know how to do this.
4. Export your database from your existing server.
 - a. You can try to chain commands on the new server to export and scp from the old one but the most common method is to simply keep 2 terminals open and follow these steps using the mysql id/pw found in your application’s env/config settings:
 - b. On the old server run:
 - i. `$ cd ~`
 - ii. `$ mysqldump -u [user] -p [database] > [filename].sql`
 - iii. You’ll be prompted to enter your user’s password.
 - c. Via FTP, connect to the old server. This will go to your user’s home directory (~). You should see the [filename].sql file there. Download this.
 - i. Via FTP, connect to the new server. Now upload the file.

Continued...

5. Import your database and set up new accounts. On the new server run:
 - a. `$ cd [directory you uploaded to in step 4.a.iii]`
 - b. `$ mysql -u [user] -p`
 - c. You'll be prompted to enter your user's password. Once entered, you'll be in the MySQL command line interface.
 - i. `mysql > CREATE DATABASE [database]`
 - ii. `mysql > CREATE USER '[user]'@'localhost' IDENTIFIED BY '[password]';`
 - iii. `mysql > GRANT ALL PRIVILEGES ON '[database]'. * TO '[user]'@'localhost';`
 - a. This gives your new user full permissions over all tables that might exist in your db
 - iv. `mysql> FLUSH PRIVILEGES;`
 - v. `mysql> exit;`
 - d. `$ mysql -u [user] -p [database] < [filename].sql`
 - i. You'll be prompted to enter your user's password.
 - e. Now update your config string in your application to the correct database name, username, password, and localhost
6. If Migrating an SSL, download from the old server (ie. via CPanel, FTP, or via your SSL provider), then upload to the new server via FTP to a path of your choosing within `"/etc/"`. If issuing a new certificate, follow your provider's instructions
 - a. **Download** our "SSL Certificates" whitepaper on our website or by contacting sales@arcanestrategies.com
7. Navigate to your HTTPD service and create a vhost. **Download** our "Creating and Enabling an Apache Vhost" whitepaper on our website or by contacting sales@arcanestrategies.com
8. Restart your HTTPD services
9. (Optional) Copy DNS entries to a new Route 53 NS instance if Changing Nameservers. Use the IP from bullet 4 in your ARecords.
10. Change DNS to new instance:
 - a. If bullet 9 completed, In your registrar, point your NS to Route 53's NS
 - b. Else just change your ARecord to the IP from bullet 4.

For migrating between servers in a single hosting provider like AWS:

1. Launch a new instance using your preferred AMI.
2. On your existing site's instance, take snapshot(s) of your site's virtual disk(s)/volume(s)
3. Connect to your instance. Download our "Connecting to an AWS Instance" whitepaper on our website or by contacting sales@arcanestrategies.com
4. Attach and mount virtual disk(s)/volume(s) to the new instance/virtual machine.
5. Download our "Creating and Mounting EBS Volumes on Linux" whitepaper on our website or by [contacting sales@arcanestrategies.com](mailto:sales@arcanestrategies.com)
6. Assign the new instance an ElasticIP
7. Follow steps 6-10 above.