**Deployment:**

Deployment is a manual process of two parts. The first is changing the database and database backup IF it is deemed to have “significant” changes. The second part is updating the code files. Neither step is complicated and there is a small window of error between updating the database and updating the code files.

We use the SVN-to-GIT inteface of Github to deploy code file changes, which are basically the git master branch.

**Database:**

As mentioned the database backup is not automatic, but if completed takes roughly 10 – 20 seconds to complete. (“cat /root/bin/mysqldump\_stairs”) All the update scripts are tested and then collated into a single script to run in one all at once.

**Code Files:**

1. Login to the production server via SSH
2. sudo to root
3. cd /home/stairs5/public\_html\_office
4. update
   1. should see updated to revision 4065
   2. and loads 4 URLs to warm up the sites
5. I usually do the SQL changes first
6. cat /root/bin/update

**Hot Standby:**

The hot standby is kept current with the production server via near real-time file and data sync processes. Configuration is the hard part because that machine is actually CPanel and any configuration changes need to be done manually. The second server is a hot standby but it needs manual intervention to come online. That's the way it was configured. Automatic failover would be hard with the machines that we currently have.

Code and uploaded files are synced every 15 minutes via “rsync”

rsync over the /home/stairs5/public\_html\_office

Database is replicated real-time.

**Failover Background:**

The site is monitored using Amazon Health check. Mack gets an alert quickly **(can a max time be determined?)** if there's connectivity loss. Manual identification of the problem (lost VM, lost connectivity for extended periods, etc.) is discovered and used to determine if a failover should be completed. If it is determined that a failover should occur, then the process is executed manually.

The failover is controlled by changes to DNS entries. The Domains & DNS are registered through GoDaddy.com. Two domain names are in 2 GoDaddy accounts, one for stairsupplies and another one for wildwoodmillwork. Mack has access to these 2 accounts. However, Mack does not have access to the third (viewrailsystems.com – also at GoDaddy.com). The TTL (time to live) on the DNS records is 10 minutes, which is the minimum available with GoDaddy.com. For a failover to work, the site will be offline for a minimum of 10 minutes.

**Servers:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Server** | **Provider** | **Type** | **Memory** | **CPUs** | **Disk** | **IP** |
| Primary | Linode | Virtual Machine | 24 GB | 8 | 380 GB SSD | 74.207.237.167 |
| Hot Standby | InMotion | Dedicated | 8 GB | 8 | 500GB (non-SSD) | 173.247.245.120 |

The current secondary is in California so ping times are not as good for Mack as the primary.

**Failover Process:**

1. If the current master is still accessible stop and disable all services:

* systemctl stop httpd coldfusion\_2016 mysql  
  systemctl disable httpd coldfusion\_2016 mysql

1. MySQL on new master:

* stop MySQL replication STOP SLAVE; RESET SLAVE ALL;   
  enable bin log:  
  log-bin=mysql-bin  
  comment out  
  read-only=1  
  restart

1. Change DNS to point to failover server for the following domains:

* sales.wildwoodmillwork.com, office.wildwoodmillwork.com  
  sales.stairsupplies.com, office.stairsupplies.com  
  sales.stairsupplies.net, office.stairsupplies.com  
  sales.viewrailsystems.com (🡨 **how is this achieved without access, or is site simply inaccessible?)**

1. Check renewal of Let's encrypt certificates

* InMotion: http://elite940.inmotionhosting.com:2086/ -> SSL/TLS -> Manage SSL Hosts  
  or  
  Linode: certbot renew  
    
  There's a cron jobs that runs every night and renews the certificates if needed  
  That's automated on the master.

**Restoration Process:**

Restoring to the primary server is again a manual process (uploaded files must be re-syncronized to the new slave, etc)