



**Open Government Platform   
Deployment Architecture**

Table of Contents

[Understanding the Deployment Architecture 3](#_Toc319663618)

[Hardware Information 3](#_Toc319663619)

[Understanding Configurations 4](#_Toc319663620)

[DNS Configuration 4](#_Toc319663621)

[Load Balancer Server Configuration 4](#_Toc319663622)

[Web Server Configuration 4](#_Toc319663623)

[File system Permissions 5](#_Toc319663624)

[APC Installation 5](#_Toc319663625)

[Passwords 5](#_Toc319663626)

[Admin Server Configuration 6](#_Toc319663627)

[SSL Configuration 6](#_Toc319663628)

[NFS Configuration 6](#_Toc319663629)

[MySQL Configuration 6](#_Toc319663630)

[MySQL Replication 6](#_Toc319663631)

[MySQL User Permissions 7](#_Toc319663632)

[SOLR Configuration 7](#_Toc319663633)

[Rsync Configuration 7](#_Toc319663634)

[Cron Configuration 7](#_Toc319663635)

[Apache Security Settings Configuration 8](#_Toc319663636)

Changes in the httpd.conf file………………………………………………………..8-9

Installation and configuration of mod\_security module………………………….9-11

MySQL failover ………………………………………………………………………………11-12

Steps followed to define FQDN names………………………………………………13

One Time Changes on the application servers (configuration file changes)………13

One time changes done on the Admin, UI, LB and DB servers…………………….14

One time changes done on the Master DB server (Create user ‘mysqlfo’ for F/O) 14

Rsync between master and slave………………………………………………………15-16

Failover scripts and location…………………………………………………………….16

Execution of F/O scripts…………………………………………………………………16

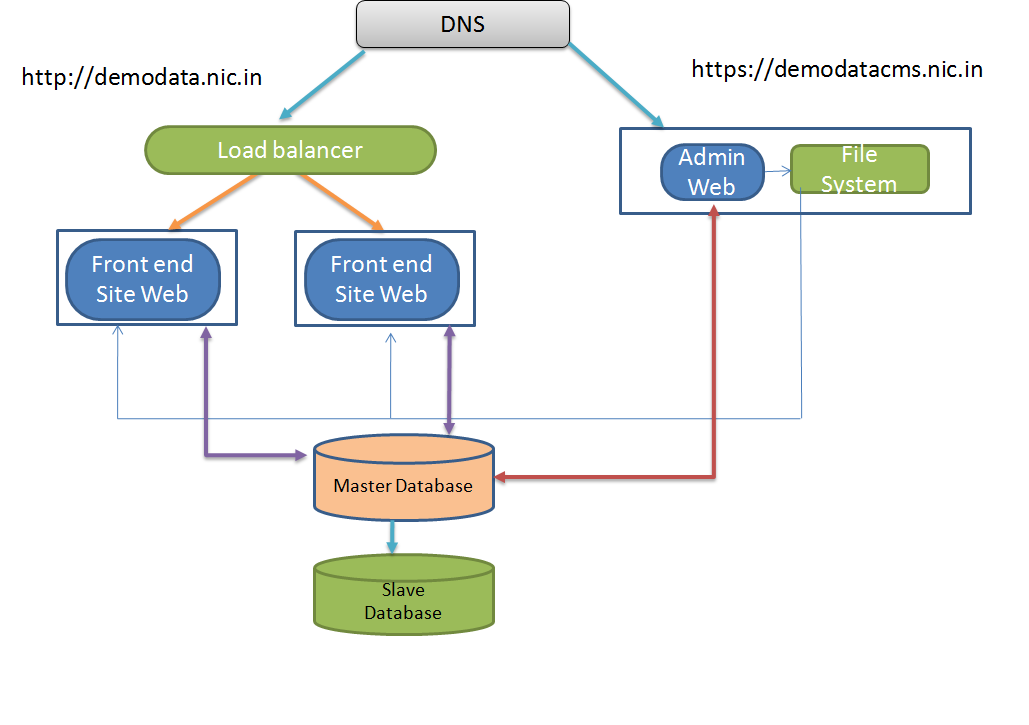
Steps for Rollback or demote old master server as a slave of promoted master….16

Replication monitoring on the slave server………………………………………………………17

Script details ………………………………………………………………………………17

# Understanding the Deployment Architecture

The following diagram illustrates the Open Government Platform Web site deployment architecture.



## Hardware Information

The following table lists the hardware required for this deployment.

|  |  |
| --- | --- |
| Hardware | Description |
| **Load Balancer Server** | This deployment requires one load balancer server. For information on the load balancer server configuration, [click here](#load). |
| **Web Server** | This deployment requires three front-end Web servers. For information on the Web server configuration, [click here](#web). |
| **Database Server** | This deployment requires two database servers. For information on the database server configuration, [click here](#db). |

# Understanding Configurations

This section provides all the configurations that you require for a successful deployment.

## DNS Configuration

The following table lists DNS-related configuration information.

|  |  |
| --- | --- |
| Item | Description |
| **Admin site name** | <https://demodatacms.nic.in> |
| **Front-end site name** | <http://demodata.nic.in> |
| **Admin site traffic** | DNS is configured to point admin site traffic to the admin server (10.153.12.183). |
| **Front End site traffic** | DNS is configured to point Front End site traffic to the load balancer server (10.153.12.182). |

## Load Balancer Server Configuration

The following table lists load balancer server-related configuration information.

|  |  |
| --- | --- |
| Item | Description |
| **Traffic Diversion** | The load balancer server 10.153.12.182 uses the round robin load balancing technique. The load balancer server is configured to divert the traffic to the following front-end Web servers:   * 10.153.12.184 * 10.153.12.185. |
| **HAProxy** | HAProxy is installed on the admin server. For information on how to install the HA proxy, [click here](http://www.webhostingtalk.com/showthread.php?t=627783). |

## Web Server Configuration

The following sections provide information about Web server-related configurations.

|  |  |
| --- | --- |
| Item | Description |
| **Admin Web Server** | 10.153.12.183 |
| **Frontend Web Server 1** | 10.153.12.184 |
| **Frontend Web Server 2** | 10.153.12.185 |

### File system Permissions

The file system permissions are based on Drupal’s standard practice for file permissions. For more information, [click here](http://drupal.org/node/244924).

A Linux group named **ogpl** was created with a user named **ogpl-user** byusing the following commands.

groupadd ogpl

adduser ogpl-user

cd /var/www/html/

chown -R ogpl-user:apache ogpl

chmod -R 750 ogpl

cd /var/www/html/ogpl/sites/default

chown -R apache:ogpl files

chmod -R 770 files

cd /var/www/html/ogpl/

chmod 640 index.php

cd /upload

chown -R apache:ogpl files

chmod -R 770 files

### APC Installation

APC is installed on all the three front-end servers by using the following commands:

yum install php-pear php-devel httpd-devel

yum install pcre-devel

pecl install apc

echo "extension=apc.so" > /etc/php.d/apc.ini

/etc/init.d/httpd restart

/etc/php.d/apc.ini

apc.shm\_segments=1

apc.shm\_size=64

### Passwords

All Linux and MySQL users have the same password as the one assigned to the **root** user.

## Admin Server Configuration

The following sections provide information about admin server-related configurations.

### SSL Configuration

The Admin site (<https://demodatacms.nic.in>) is configured with OpenSSL, which is running on port **443**. For information on how to configure OpenSSL, [click here](http://wiki.centos.org/HowTos/Https).

### NFS Configuration

The NFS directory is configured on the Admin Web server.

The NFS file system is mounted on the **upload** directory, on both front-end Web servers.

**Note:** The **upload** directory is used as the Drupal file system directory.

## MySQL Configuration

The following sections provide information about MySQL-related configurations.

### MySQL Replication

The following table lists MySQL replication-related configuration information.

|  |  |
| --- | --- |
| Item | Description |
| **Master Server** | 10.153.12.186 |
| **Slave Server** | 10.153.12.187 |
| **Max\_allowed\_packet** | Set to 64 MB.  Set in the **/etc/my.cnf** file for both the master and the slave server. |
| **Replication on Slave Server** | The following lines are added to the **/etc/my.cnf** file so that the following tables are not replicated:  replicate-wild-ignore-table=ogpl%.cache%  replicate-wild-ignore-table=ogpl%.watchdog% |

For information on how to setup replication, see <http://www.developertutorials.com/tutorials/mysql-tutorials/implementing-high-availability-in-mysql-8-01-14-952/> or <http://aciddrop.com/2008/01/10/step-by-step-how-to-setup-mysql-database-replication/>.

### MySQL User Permissions

The following table lists MySQL replication-related configuration information.

|  |  |
| --- | --- |
| Item | Description |
| **Database name** | **ogpl** |
| **MySQL Users** | The following users are created:   * **root** * **ogpl** This user has the write access to all tables on the Admin Web server. * **ogpl\_web** This user has the write access to some tables while accessing from both front end Web servers. The Drupal **settings.php** file uses the **ogpl\_web** user for the connection. |

For more information on user permissions, double-click the following icons:



## SOLR Configuration

SOLR lucene search service is configured on the slave database server. This library is installed on the /opt/apache-solr-3.5.0 directory. SOLR is running with tomcat /opt/tomcat.

## Rsync Configuration

Rsync is configured on the Admin Web server and the script is available at **/home/sync\_code.sh**. After the code is deployed on the Admin Web server, execute **sh /home/sync\_code.sh**. This deploys the code on both frontend Web servers, except for a few files that are mentioned in **/home/exclude\_list.txt**.

## Cron Configuration

The Drupal cron script is configured to execute on the Admin Web server. On other front end servers, the **exit** command is added to **/var/www/html/ogpl/cron.php**. This ensures that the cron script is executed only from the Admin Web server.

## Apache Security Settings Configuration

Changes in the httpd.conf file

The Apache web server is a crucial part of the website infrastructure. It has a number of built in features that can improve your website resistance to attacks. In this section we have covered various ways to secure apache web server to mitigate the attacks.

Following table shows the various parameters from the **httpd.conf** file, their current values and the values we have updated in the httpd.conf file.

We have updated httpd.conf file with following values on 2 FE servers (10.153.12.184, 10.153.12.185) and 1 Admin Server (10.153.12.183)

Below are the steps we have followed to update these values.

1. Login with root user on 10.153.12..183
2. cd /etc/httpd/conf
3. cp httpd.conf httpd.conf\_13mar12
4. vi httpd.conf
5. Updated below parameters to the suggested values
6. Restarted apache service (service httpd restart)

Note: - Same steps are followed on all web servers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **httpd.conf changes** | | | | | |
| **NO** | **Parameter** | **Current Value** | **Suggested Value** | **Importance** | **Benefits for web site** |
|  |  |  |  |  |  |
| 1 | ServerSignature | on | off | The **ServerSignature off** parameter will avoid showing information about webserver version and it' s port number on the custom or default error page of the site | 1. It will stop sending information regarding apache version and port number to the internet |
| 2 | ServerTokens | OS | Prod | **ServerTokens prod** parameter will avoid showing information about full OS name and version to be displayed on the internet. Hackers will not get any information about OS and apache version installed on the system. | 1. Hackers will not get information about OS version and server technology |
| 3 | Disable unwanted modules  1.dav\_module\_fs 2. dav\_module 3. mod\_userdir | Enabled | Disabled | 1/2. WebDAV is a file access protocol created over HTTP protocol. It allows you to upload and download files, and change file contents from the website. This service is required only in very rare cases. From our experience, this feature was only required to run SVN server (link). Make sure that WebDAV is disabled in production websites 3. Avoid Mapping of requests to user-specific directories. i.e ~username in URL will get translated to a directory in the server | 1. Unwanted modules will get disabled |
| 4 | Directory indexing | Enabled (options Indexes Followsymlinks) | Disabled (options None) | 1. It will disallow to list the contents of directory | 1. Disable directory indexing in browser. If any hackers wants to list the contents of directory he/she will get error of forbidden. |
| 5 | Timeout | 120 | 45 | Help to mitigate the potential effects of a denial of service attack. | 1. Help to mitigate DDOS attack |
| 6 | Tuning parameters for apache startservers MinSpareServers MaxspareServers ServerLimit MaxClients MaxRequetsPerchield | 8 5 20 256 256 4000 | 20 20 50 300 300 5000 | 1. Avoid large number of request being in pending state. Note: - one can readjust these values based on the actual usage of apache threads. | 1. It will avoid incoming request being in queue |
| 7 |  |  |  |  |  |

Installation and Configuration of mod\_security module

ModSecurity is a web application firewall (WAF). WAF is deployed to establish an increased external security layer to detect and/or prevent attacks before they reach web applications. ModSecurity provides protection from a range of attacks against web applications and allows for HTTP traffic monitoring and real-time analysis with little or no changes to existing infrastructure.

Advantages of mod\_security

1. HTTP Traffic Logging
2. Real-Time Monitoring and Attack Detection
3. Attack Prevention and Just-in-time Patching
4. Embedded-mode Deployment

Installation of mod\_security

1. Logged in with root user on admin server (10.153.12.183)
2. Gave command to install mod\_security with all its dependencies. Following package were got installed.
3. Yum install mod\_security

======================================================================================

Package Arch Version Repository Size

======================================================================================

Installing:

mod\_security x86\_64 2.5.12-3.el5 epel 1.0 M

1. Above installation has created one file mod\_security.conf under /etc/httpd/conf.d and one directory modsecurity.d under /etc/httpd/
2. Modsecurity.d directory contains default and local rules which will get apply to the website.

Configuration of mod\_security

1. Important configuration files for mod\_security are ***mod\_security.conf*** and ***modsecurity\_crs\_10\_config.conf***
2. Edited file above two files and changed the value for “SecRuleEngine” parameter as given below.
   1. Vi /etc/httpd/conf.d/mod\_security.conf
      1. SecRuleEngine DetectionOnly
   2. Vi /etc/httpd/modsecurity.d/modsecurity\_crs\_10\_config.conf
      1. SecRuleEngine DetectionOnly
3. Restarted apache service
   1. /etc/init.d/httpd restart

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Important Files | /etc/httpd/conf.d/mod\_security.conf /etc/httpd/modsecurity.d/modsecurity\_crs\_10\_config.conf | | | | |
| **NO** | **Parameter** | **Current Value** | **Suggested Value** | **Importance** | **Benefits for web site** |
| 1 | SecRuleEngine | on | DetectionOnly | 1. It will only generate triggers , it will not directly block the traffic for the site | Mod\_securty in DetectionOnly mode minimize any disruptions to traffic until you get a handle on how your configs/rules will respond to your traffic.  This setting allows SecRules to trigger events but not take any disruptive actions |
| 2 | SecRequestBodyLimit | 131072 | 5242880 | It will help to upload audio/video files of size more than 5 MB on admin server | It will allow to upload audio/video files of more than 5 MB on admin server |
| Note: - We have installed mod\_security on the system and change its value to "DetectionOnly". This setting will process all default rules however will not disturb the current site traffic. It will log the suspected data in the audit.log file | | | | | |

Note: - Same steps are followed on all web servers

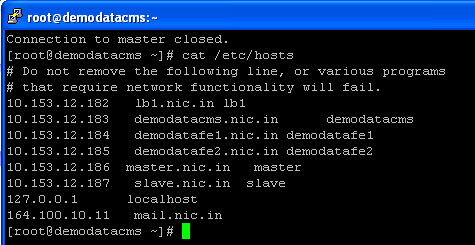
## MySQL Failover Setup

For MySQL failover we are following the script based approach. 10.153.12.183 (Admin) server is assumed as central monitoring server. MySQL-monitoring script will run on this system and it will do remote failover to the slave server if it finds that MySQL Master is down.

We are using FQDN names of all systems in the MySQL F/O scripts. Following table shows the FQDN names of all systems

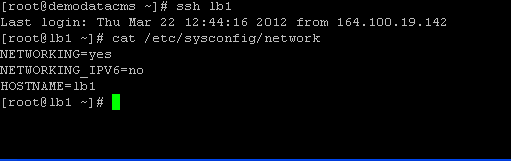
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine name** | **IP address** | **FQDN (Yes/No)** | **Current Name** | **Suggested FQDN Name** |
| Load Balancer | 10.153.12.182 | No | localhost.localdomain | lb1.nic.in |
| Admin server | 10.153.12.183 | Yes | demodatacms.nic.in |  |
| Frontend 1 | 10.153.12.184 | Yes | demodata.nic.in | demodatafe1.nic.in |
| Frontend 1 | 10.153.12.185 | Yes | localhost.localdomain | demodatafe2.nic.in |
| Mysql Master | 10.153.12.186 | No | localhost.localdomain | master.nic.in |
| Mysql slave | 10.153.12.187 | No | localhost.localdomain | Slave.nic.in |

We have added entries for all hostnames in the /etc/hosts files on all servers. Following are the /etc/hosts entries from one of the systems.

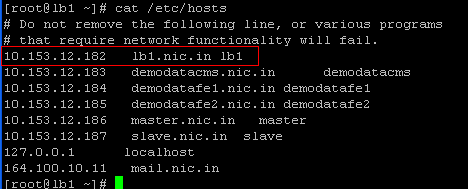


**Steps followed to define FQDN names**

1. Set the hostname kernel parameter and change the name of system from “localhost.localdomain” to “lb1”
   1. Sysctl kernel.hostname=lb1
2. Edited the file /etc/sysconfig/network and changed the hostname from “localhost.localdomain” to “lb1”



1. Edited /etc/hosts file on the system and added the FQDN entry for the system



1. Ran the command “hostname” on the system. It has shown the name “lb1” as the hostname

***Note: - We have followed the same above steps on the rest of the systems to assign the hostnames and FQDN names.***

**One Time Changes on the application servers (configuration file changes)**

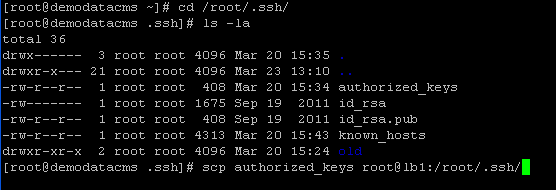
Updated configuration file “settings.php” and replaced hard coded IP address for DB server to the hostname of DB server. This change will help application to point to new master server once F/O gets done.

**One time changes done on the Admin, UI, LB and DB servers:-**

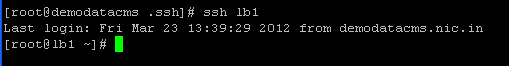
In the current setup Dev team uses admin server (10.153.12.183) to sync data with both UI servers; (10.153.12.184, 10.153.12.185)

We found that private/public keys are generated on 10.153.12.183 in /root/.ssh and copied as ***authorized\_keys*** on 10.153.12.184, 10.153.12.185.

During F/O monitoring system should get the password less login to all the systems to do required changes in system so we copied /root/.ssh/authorized\_key file from 10.153.12.183 system to rest of the systems in /roo/.ssh folder.



Log in to any of the system and confirmed that it should not ask for the password while login from 10.153.12.183



**One time changes done on the Master DB server (Create user ‘mysqlfo’ for F/O)**

Master and Slave db servers have ogpl, ogpl\_web and root users however these users has limited rights on the database. We have created a “mysqlfo” user with full access to be used in the F/O script.

Following commands we ran on the master server to create user and give rights to it.

create user 'mysqlfo'@'localhost' identified by 'default';

grant all privileges on \*.\* to 'mysqlfo'@'localhost' with grant option;

create user 'mysqlfo'@'%' identified by 'default';

grant all privileges on \*.\* to 'mysqlfo'@'%' with grant option;

**Rsync between master and slave**

During F/O through the script we are using file /etc/my\_promote.cnf from the slave server to get change as a /etc/my.cnf on the server so that during promotion of slave to master it will reflect the changes exactly as there on old master server.

We have created a ssh key on the master server and copied on the slave so that they will communicate with each other without password. We are using rsync utility to update the changes in the /etc/my.cnf file on master server to the /etc/my\_promote.cnf file on slave server.

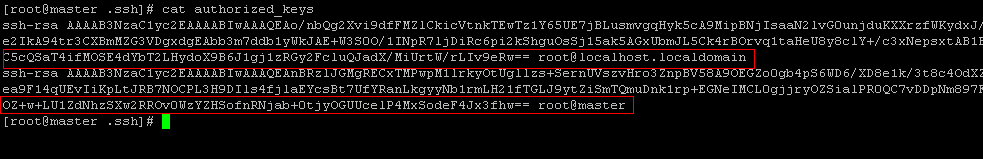
We will schedule a cron for that which will sync file /etc/my.cnf with the /etc/my\_promote.cnf on the slave.

Steps we executed on master and slave

1. On master server ran the command to generate public/private key
   1. ssh-keygen -t rsa
   2. it has created two files id\_rsa and id\_rsa.pub
   3. we already have a public key from monitoring server (10.153.12.183) on the master and slave servers.
   4. Append new public key from master to the authorized\_key file on the master server
      1. cat id\_rsa.pub >> authorized\_keys



* 1. Check authorized\_keys file on the system. It should show two public keys. One for monitoring server (10.153.12.183) and one for local system (10.153.12.186)



* 1. Copy the same key from master server to slave server.
     1. [root@master .ssh]# scp authorized\_keys [root@slave:/root/.ssh](mailto:root@slave:/root/.ssh)
  2. Make sure that you can now connect without password from master to slave server also.

Now with the rsync utility we can sync the file /etc/my.cnf with the /etc/my\_promote,cnf on the slave server.

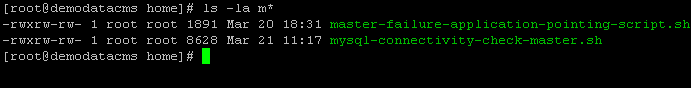
rsync -v /etc/my.cnf root@slave:/etc/my\_promote.cnf

Note: - .1. Do not change name of file /etc/my\_promote.cnf on the slave server as it will affect the F/O process.

2. Master and slave servers will have public keys for 10.153.12.183 and 10.153.12.186

**Failover scripts and location**

Failover scripts are located at /home on 10.153.1.2183 server

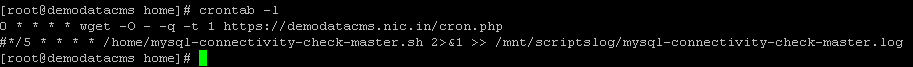
****

|  |  |
| --- | --- |
| Script Name | Description |
| /home/mysql-connectivity-check-master.sh | This script will check the availability of MySQL service on port 3306 on master server. If it finds that MySQL service is not responding it will try to start it. If script is not able to start the service it will do the F/O on the slave severer. |
| /home/master-failure-application-pointing-script.sh | This script will get call from the mysql-connectivity-check-master.sh script. Once the F/O is done this script will change the /etc/hosts files on the application and LB servers so that application should get the IP address of new master server and it will point to new server |

**Execution of F/O scripts**

F/O scripts will get executed as cron job on the 10.153.12.183 server

\*/5 \* \* \* \* /home/mysql-connectivity-check-master.sh 2>&1 >> /mnt/scriptslog/mysql-connectivity-check-master.log



**Steps for Rollback or demote old master server as a slave of promoted master**

These steps are mentioned in a separate document “mysql-rollback.docx”

## Replication monitoring on the Slave Server

We have created a script to monitor replication service on the slave server. It will check for the replication service and send email alerts to the respective persons if replication service has stopped.

|  |  |
| --- | --- |
| Script Name | Description |
| /home/replication\_monitoring\_production\_slave.sh | This script will monitor the status of Slave\_IO\_Running and Slave\_SQL\_Running from the slave server. If it finds that any of them is not running it will send alert to the respective persons to take action on it. |