

# ????????? ??? 20.04 ?? 24.04 - steps

?? ?????? ??? ?????? ?????? ??? ?????? ?? ????????????

1. database server & main app για διαχειριση mydata δεδομενων
2. ergani sync εργασιες (<http://localhost:7000>, /workers)
3. salary εργασίες (<http://localhost:7001>, /salary)
4. notification για ληξη ssl, λιγος χωρος κτλπ (server monitoring)
- 5.

????????? ????? ??????? 178.105.43.239

1. install mysql [X]
2. secure mysql και ρυθμιση mysql παραμετρων να δειχνουμε το datadir στον μεγαλο σκληρο (2TB) [X]
3. στησιμο apache2 με ρυθμισεις απο το προηγουμενο σερβερ
  1. Ο προηγουμενος σερβερ ετρεχε prefork με τις παρακατω ρυθμισεις <IfModule mpm\_prefork\_module>  
ServerLimit 250  
StartServers 10  
MinSpareServers 5  
MaxSpareServers 10  
MaxRequestWorkers 150  
MaxConnectionsPerChild 0  
MaxConnectionsPerChild 10000  
</IfModule>
  2. Ο καινουργιος σερβερ δε θα τρεχει prefork αλλα να εχουμε υποψιν τα ορια που μπηκαν [X]
4. στησιμο SSL
5. να ξεκινησουνε τα services του 7000, 7001[X]
- 6. ρυθμιση σωστης ωρας [X]**
7. Στησιμο modsecurity [OPTIONAL] (ο προηγουμενος δεν ειχε)
8. install gunicorn [X]
9. install php [X]
10. install php modules used in server [X]
11. install cronjobs from previous server [X] **EINAI SE SXOLIA! (και εχουν επισης php8.2 αντι για 8.3)**
12. install REDIS server [X]

# 1. Step-by-Step Configuration

Πρέπει πρώτα να αλλάξουμε τον main server για να έχει bind-address: 0.0.0.0 ώστε να μπορεί να γίνει το replication. Επίσης για να γίνει self-healing το replcaiton είναι προτιμότερο να τρέχει με GTIDs:

```
# Enable GTID mode
gtid_mode = ON
```

```
# Ensure only GTID-safe statements are allowed
enforce_gtid_consistency = ON
```

```
# Required for GTIDs to work with replication
```

```
log_bin = mysql-bin
```

```
log_slave_updates = ON
```

Ο λόγος που είναι καλύτερα τα GTIDs (<https://gemini.google.com/share/aac5bf6e69e6>) επίσης δεν χρειάζεται να βαζουμε καθολου τα replication positions που κάνει τη διαδικασία πιο ασφαλές

## Phase A: Prepare the Donor (Primary)

Log into your **Primary** server and run:

SQL

```
-- Install the plugin
INSTALL PLUGIN clone SONAME 'mysql_clone.so';

-- Create a replication user with cloning permissions
CREATE USER 'clone_user'@'%' IDENTIFIED BY 'password';
GRANT BACKUP_ADMIN ON *.* TO 'clone_user'@'%';
```

## Phase B: Prepare the Recipient (Replica)

Log into your **New Replica** server and run:

SQL



```
-- Install the plugin
INSTALL PLUGIN clone SONAME 'mysql_clone.so';

-- Allow the recipient to be overwritten by the donor
SET GLOBAL clone_valid_donor_list = 'primary-ip-address:3306';
```

## Phase C: Execute the Clone

On the **Recipient** server, trigger the data transfer. **Warning:** This will delete all existing data on the recipient and restart the MySQL service automatically after completion.

SQL



```
CLONE INSTANCE FROM 'clone_user'@'primary-ip-address':3306 IDENTIFIED BY 'password';
```

---

## 3. Completing the Replication

The beauty of the Clone plugin is that it automatically captures the **Binary Log** coordinates (GTID or filename/position) during the transfer.

Once the recipient restarts, log back in and check the status:

SQL



```
SELECT BINLOG_FILE, BINLOG_POS FROM performance_schema.clone_status;
```

# Option 1: If using GTIDs (Recommended)

If your primary uses GTIDs, replication might start automatically or simply require:

SQL

■ ■

```
CHANGE REPLICATION SOURCE TO
    SOURCE_HOST='primary-ip-address',
    SOURCE_USER='repl_user',
    SOURCE_PASSWORD='password',
    SOURCE_AUTO_POSITION=1;
START REPLICA;
```

# Option 2: If using Binary Log Position

SQL

■ ■

```
-- Use the coordinates found in performance_schema.clone_status
CHANGE REPLICATION SOURCE TO
    SOURCE_HOST='primary-ip-address',
    SOURCE_USER='repl_user',
    SOURCE_PASSWORD='password',
    SOURCE_LOG_FILE='binlog.00000X',
    SOURCE_LOG_POS=123;
START REPLICA;
```

?????? ?????? ?? replication:

Μολις γινει το replication βρισκομαστε στο τελευταιο βημα θα πρεπει να μεταφερουμε το domain απο το server A, στο B. Σε αυτο το βημα θα πρεπει ο master να γινει READ-ONLY και ο REPLICATION να γινει ο νεος μαστερ.

1. Βαζουμε το master σε READ-ONLY και το REPLICATION γινεται MASTER
2. κανουμε point ολα τα σημεια που συνδεονται στη βαση του mydata.semantic.gr στο 10.0.0.13 που ειναι η εσωτερικη IP του νεου σερβερ
3. Αλλαζουμε το A record του παλιου σερβερ στο νεο σερβερ. Σε αυτη τη φαση καποιοι κοιτανε το παλιο mydata.semantic.gr και καποιοι το καινουργιο. Εφосον δειχνουμε απο το παλιο στη καινουργια βαση ειναι ακριβως το ιδιο.

4. Περιμενουμε 7 μερες και μετα κλεινουμε τελειως το παλιο σερβερ (ολοι θα βλεπουν πλεον ουτως η αλλος το καινουργιο σερβερ)

---

Revision #8

Created 2026-04-30 08:14:32 UTC by Admin

Updated 2026-05-04 09:59:39 UTC by Admin