ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Αυτοματοποίηση Επίβλεψης και Αντιγράφων Ασφαλείας για Συστήματα Βάσεων Δεδομένων MySQL και MariaDB»

Του φοιτητή
Χρήστου-Παναγιώτη Κάκτση
Αρ. Μητρώου: 154455

Επιβλέπων
Αντώνης Σιδηρόπουλος
Επίκουρος Καθηγητής

Ημερομηνία 24-08-2022

Τίτλος Δ.Ε. Αυτοματοποίηση Επίβλεψης και Αντιγράφων Ασφαλείας για Συστήματα Βάσεων Δεδομένων MySQL και MariaDB Κωδικός Δ.Ε. 21343

Ονοματεπώνυμο φοιτητή/των Χρήστος – Παναγιώτης Κάκτσης

Ονοματεπώνυμο εισηγητή Αντώνης Σιδηρόπουλος

Ημερομηνία ανάληψης Δ.Ε. 13-10-2021

Ημερομηνία περάτωσης Δ.Ε. …

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Πρόλογος

Στη σημερινή εποχή όλες οι εφαρμογές βασίζονται σέ ένα back end περιβάλλον. Το back end είναι ένας server ο οποίος φιλοξενεί μία βάση δεδομένων η οποία είναι υπεύθυνη για την αποθήκευση των δεδομένων. Για να μπορέσει κάποιος να διαχειριστεί την βάση θα πρέπει να έχει εγκαταστήσει στον server ένα σύστημα διαχείρισης βάσης δεδομένων (DBMS) όπως το MySQL ή το MariaDB.

Θα ήταν πολύ χρήσιμο, ένας που η κύρια ασχολία του είναι η ανάπτυξη εφαρμογών, να γνωρίζει πώς διαχειρίζονται τα συστήματα αυτά και να μπορεί να καταλάβει σε βάθος την λειτουργεία τους.

Μέσα από αυτή την πτυχιακή θα μπορέσω να εγκαταστήσω τα συστήματα MySQL και MariaDB, σε ένα Virtual Machine περιβάλλον, να μάθω πώς λειτουργούν και με τις απαραίτητες γνώσεις που θα συγκεντρώσω θα δώσω λύση στο πρόβλημα.

Πιστεύω πως στο τέλος θα έχω αποκτήσει εμπειρίες που θα ενισχύσουν την πορεία μου ως προγραμματιστή εφαρμογών.

Περίληψη

Σκοπός της πτυχιακής είναι η ημι αυτοματοποιημένη διαχείριση υπηρεσιών DBMS (της οικογένειας mysql).Πιο συγκεκριμένα η πτυχιακή θα υλοποιηθεί με το παράδειγμα των βάσεων δεδομένων που είναι εγκατεστημένοι στο τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων. Αναλυτικότερα υπάρχει μια πληθώρα από Η/Υ και υπηρεσίες DBMS που τρέχουν στο τμήμα. Οι HY/servers μπορεί να είναι ετερογενείς, δηλαδή να εκτελούν διαφορετικά λειτουργικά συστήματα. Επίσης και οι DBMS υπηρεσίες είναι ετερογενείς. Υπάρχουν εγκατεστημένοι mysql servers και mariadb servers διαφόρων εκδόσεων. Το ζητούμενο είναι η δημιουργία μιας υπηρεσίας “ομπρέλα” για την διαχείριση και την παρακολούθηση των υπηρεσιών αυτών. Στην διαχείριση περιλαμβάνεται και το συστηματικό αυτοματοποιημένο backup των δεδομένων. Ο τρόπος με τον οποίο μπορεί να υλοποιηθεί το θέμα παρακολούθησης είναι με την χρήση ενός εργαλείου που ονομάζεται Zabbix. Το Zabbix μπορεί να εγκατασταθεί στο κύριο τερματικό που έχει κληθεί να κάνει την διαδικασία παρακολούθησης. Το κομμάτι του ημι αυτοματοποιημένου backup μπορεί να υλοποιηθεί με την δημιουργία script το οποίο συνδέεται μέσω ssh σε κάθε έναν server και με τη χρήση του εργαλείου mydumper, το οποίο είναι υπεύθυνο για δημιουργία backup σε mysql - mariaDB server, ολοκληρώνεται η διαδικασία. Για να είναι πιο ασφαλής, άμεση η σύνδεση ssh γίνεται με την χρήση private public key. Ο χρήστης, που έχει δημιουργηθεί σε κάθε database server, κάνει σύνδεση με το socket pluggable authentication, που του επιτρέπει να κάνει σύνδεση χωρίς username και password.
Το θέμα της πτυχιακής δοκιμάστηκε και υλοποιήθηκε σε Virtual machine περιβάλλον.

«Automation of monitoring and backing-up for MySQL and MariaDB DBMS»

«Christos – Panagiotis Kaktsis»

Abstract

The purpose of the thesis is the semi-automated management of DBMS services (of the mysql family). More specifically, the thesis will be implemented with the example of the databases installed in the Department of Information Technology and Electronic Systems Engineering. Specifically there is a multitude of PCs and DBMS services running in the department. PCs/servers can be heterogeneous, i.e. running different operating systems. Also, DBMS services are heterogeneous. There are installed MySQL servers and mariadb servers of various versions. The aim is to create an "umbrella" service for the management and monitoring of these services. The management also includes the systematic automated backup of the data. The way in which the monitoring issue can be implemented is by using a tool called Zabbix. Zabbix can be installed on the main terminal that has been called to do the monitoring process. The part of the semi-automated backup can be implemented by creating a script that connects via ssh to each server and by using the mydumper tool, which is responsible for creating a backup on MySQL - MariaDB server, the process is completed. To be more secure, the direct ssh connection is made using a private public key. The user, created on each database server, makes a connection with the pluggable authentication socket, which allows him to make a connection without username and password.
The subject of the thesis was tested and implemented in a Virtual machine environment.

Ευχαριστίες

Ευχαριστώ πολύ τον κύριο Αντώνη Σιδηρόπουλο που με βοήθησε σε όλη την πορεία της πτυχιακής μου και την σύζυγό μου που έκανε υπομονή όλον αυτό τον καιρό μέχρι να ολοκληρώσω τις σπουδές μου.

Περιεχόμενα

Πρόλογος **3**

Περίληψη **4**

Abstract **5**

Ευχαριστίες **6**

Περιεχόμενα **7**

DBMS **1**

- Εισαγωγή 1
- Τι είναι η Βάση δεδομένων (Database) 1
- Τι είναι ένα DBMS 1
- Mysql 1
  - Τι είναι η Mysql 1
  - Ιστορία Mysql 2
- MariaDB 2
  - Τι είναι η MariaDB 2
  - Ιστορία 2
- Τρόπος σύνδεσης 2
- Pluggable Authentication 4
- Επίλογος 5

Backup Server **6**

- Αντίγραφο ασφαλείας 6
- Backup types 6
  - Physical Vs Logical Backups 6
  - Online vs Offline Backups 6
  - Τοπικά vs Απομακρυσμένα backups 7
  - Snapshot backups 7
  - Backup Schedulling 7
- Εργαλεία αντιγράφου ασφαλείας 7
  - Mysqldump 7
  - MyDumper 8
- Επίλογος 8

Monitor Server **9**

- Εισαγωγή 9

Τι είναι το Monitor 9
Μετρήσεις MySQL 9
Μεταβλητές κατάστασης Server 9
Performance Schema 10
Sys Schema 11
Performance Monitoring Tools 11
MySQL Workbench 11
Επίλογος 11
Backup Servers Η λύση στο πρόβλημα 12
Εισαγωγή 12
Shell Script 12
Τι είναι το shell script 12
Παράμετρος στο script 12
Η συνθήκη if και η εντολή test 13
Βρόχος while do και ανάγνωση γραμμής αρχείου 13
Η λίστα με τους servers 13
Απομακρυσμένη σύνδεση με SSH 14
Σύνδεση στη mysql χωρίς κωδικό 15
Δημιουργία Shell Script 15
To Script 15
CronTab και προγραμματισμένη εκτέλεση script 18
Επίλογος 18
Παρακολούθηση Συστημάτων Η λύση στο πρόβλημα 19
Εισαγωγή 19
Zabbix 19
Αρχιτεκτονική Zabbix 19
Zabbix Server 20
Zabbix Agent 20
Επίλογος 20
Συμπεράσματα και προτάσεις βελτίωσης 21
ΒΙΒΛΙΟΓΡΑΦΙΑ 22

Κεφάλαιο 1ο: DBMS

1.1 Εισαγωγή

Στον κόσμο της τεχνολογίας, η ανταλλαγή δεδομένων και πληροφορίας είναι κύριο συστατικό για την ύπαρξή της. Καθημερινά χρησιμοποιούμε προγράμματα και εφαρμογές για να επικοινωνήσουμε μεταξύ μας, να αποθηκεύσουμε και να ανακτήσουμε σημαντικές πληροφορίες. Όλος αυτός ο όγκος δεδομένων είναι σημαντικό, να αποθηκευτεί σε κάποιο χώρο ο οποίος παρέχει ασφάλεια και χειρίζεται τα δεδομένα αυτά με οργανωμένο τρόπο, ώστε να μπορούμε να τα επεξεργαζόμαστε με ευκολία.

1.2 Τι είναι η Βάση δεδομένων (Database)

Μια βάση δεδομένων είναι μια οργανωμένη συλλογή δομημένων πληροφοριών ή δεδομένων, που συνήθως αποθηκεύονται ηλεκτρονικά σε ένα υπολογιστικό σύστημα. Μπορεί να είναι οτιδήποτε, από μια απλή λίστα αγορών μέχρι μια συλλογή εικόνων ή τις τεράστιες ποσότητες πληροφοριών σε ένα εταιρικό δίκτυο. Μια βάση δεδομένων ελέγχεται συνήθως από ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Μαζί, τα δεδομένα και το DBMS, μαζί με τις εφαρμογές που σχετίζονται με αυτά, αναφέρονται ως σύστημα βάσης δεδομένων, συχνά συντομευμένο σε απλή βάση δεδομένων[1].
Τα δεδομένα στους πιο συνηθισμένους τύπους βάσεων δεδομένων που λειτουργούν σήμερα μοντελοποιούνται συνήθως σε γραμμές και στήλες σε μια σειρά πινάκων για να καταστήσουν αποτελεσματική την επεξεργασία και την αναζήτηση δεδομένων. Στη συνέχεια, τα δεδομένα μπορούν να είναι εύκολα, προσβάσιμα, στη διαχείριση, στη τροποποίηση, στην ενημέρωση, στον έλεγχο και στην οργάνωση. Οι περισσότερες βάσεις δεδομένων χρησιμοποιούν την γλώσσα structured query language (SQL) για τη σύνταξη και την αναζήτηση δεδομένων[1].

1.3 Τι είναι ένα DBMS

Μια βάση δεδομένων συνήθως απαιτεί ένα ολοκληρωμένο πρόγραμμα λογισμικού βάσης δεδομένων, γνωστό ως σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Ένα DBMS χρησιμεύει ως διεπαφή μεταξύ της βάσης δεδομένων και των τελικών χρηστών ή προγραμμάτων, επιτρέποντας στους χρήστες να ανακτούν, να ενημερώνουν και να διαχειρίζονται τον τρόπο οργάνωσης και βελτιστοποίησης των πληροφοριών. Ένα DBMS διευκολύνει επίσης την επίβλεψη και τον έλεγχο των βάσεων δεδομένων, επιτρέποντας μια ποικιλία διοικητικών λειτουργιών όπως παρακολούθηση απόδοσης, συντονισμός και δημιουργία αντιγράφων ασφαλείας και ανάκτηση[1].

1.4 Mysql

1.4.1 Τι είναι η Mysql

Η MySQL είναι ένα open source σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Σχεδιάστηκε και βελτιστοποιήθηκε για εφαρμογές web και μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα. Καθώς εμφανίστηκαν νέες και διαφορετικές απαιτήσεις με το Διαδίκτυο, η MySQL έγινε η επιλογή για διαδικτυακές εφαρμογές . Επειδή έχει σχεδιαστεί για να επεξεργάζεται εκατομμύρια ερωτήματα και χιλιάδες συναλλαγές, η MySQL είναι μια δημοφιλής επιλογή για επιχειρήσεις ηλεκτρονικού εμπορίου που πρέπει να διαχειρίζονται πολλαπλές μεταφορές χρημάτων. Η ευελιξία κατ' απαίτηση είναι το κύριο χαρακτηριστικό της MySQL[1].
Η MySQL είναι το DBMS πίσω από μερικούς από τους κορυφαίους ιστότοπους και εφαρμογές, όπως Airbnb, Uber, LinkedIn, Facebook, Twitter και YouTube.

1.4.2 Ιστορία Mysql

Οι ρίζες της MySQL ξεκινούν το 1979, με το εργαλείο βάσης δεδομένων UNIREG που δημιουργήθηκε από τον Michael "Monty" Widenius για τη σουηδική εταιρεία TcX. Το 1994, το TcX άρχισε να ψάχνει για ένα RDBMS με διεπαφή SQL για χρήση στην ανάπτυξη εφαρμογών Web. όλοι οι εμπορικοί διακομιστές που δοκιμάστηκαν ήταν υπερβολικά αργοί για τα μεγάλα δεδομένα του TcX και η δωρεάν διαθέσιμη mSQL δεν είχε τα χαρακτηριστικά που απαιτούσε το TcX. Κατά συνέπεια, ο Monty άρχισε να αναπτύσσει έναν νέο διακομιστή.
Το 1995, ο David Axmark της Detron HB άρχισε να πιέζει ώστε το TcX να κυκλοφορήσει τη MySQL στο Διαδίκτυο. Η MySQL 3.11.1 κυκλοφόρησε στον κόσμο το 1996 για Linux και Solaris. Η εταιρεία MySQL AB δημιουργήθηκε για να παρέχει διανομή της MySQL και να προσφέρει εμπορικές υπηρεσίες. Το 2008, η Sun Microsystems εξαγόρασε τη MySQL AB, και το 2010, η Oracle εξαγόρασε τη Sun. Σήμερα, η MySQL είναι διαθέσιμη τόσο σε δυαδικό όσο και σε source μορφή και λειτουργεί σε πολλές ακόμη πλατφόρμες[2].

1.5 MariaDB

1.5.1 Τι είναι η MariaDB

Η MariaDB είναι και αυτή ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, η οποία είναι βασισμένη στην Mysql[3].

1.5.2 Ιστορία

Το 2008, η MySQL εξαγοράστηκε από τη Sun Microsystems, η οποία με τη σειρά της εξαγοράστηκε από την Oracle Corporation το 2010. Ενώ η αρχική εξαγορά από τη Sun θεωρήθηκε, από πολλούς στην κοινότητα της MySQL, ως ακριβώς αυτό που χρειαζόταν, αυτό το συναίσθημα δεν κράτησε και η μετέπειτα εξαγορά από την Oracle δυστυχώς αντιμετωπίστηκε με πολύ χαμηλότερες προσδοκίες.
Πολλοί από τους προγραμματιστές της MySQL εγκατέλειψαν τη Sun και την Oracle για να εργαστούν σε νέα έργα. Ανάμεσά τους ήταν και ο Michael “Monty” Widenius.
Ο Monty και η ομάδα του δημιούργησαν ένα fork του πηγαίου κώδικα MySQL και το ονόμασαν MariaDB. Καθώς η MariaDB βασίζεται στη MySQL, μοιράζεται τα οφέλη της Mysql που αναφέρθηκαν προηγουμένως.[3]

1.6 Τρόπος σύνδεσης

Υπάρχουν διάφορες μέθοδοι αλληλεπίδρασης με έναν διακομιστή MySQL ή MariaDB για ανάπτυξη ή επεξεργασίας μιας βάσης δεδομένων. Ένα πρόγραμμα που συνδέεται με τον server είναι γνωστό ως MySQL Client.
Με τον client mysql, μπορούμε να αλληλοεπιδράσουμε με τον διακομιστή MySQL ή MariaDB από τη γραμμή εντολών ή σε ένα γραφικό περιβάλλον. Η μέθοδος γραμμής εντολών επιτρέπει την άμεση αλληλεπίδραση με τον διακομιστή. Επιτρέπει επίσης την εισαγωγή εντολών Mysql σε script.
Εάν η MySQL ή η MariaDB έχουν εγκατασταθεί σωστά στον server, η mysql θα πρέπει να είναι διαθέσιμη για χρήση. Υποθέτοντας ότι όλα λειτουργούν, θα χρειαστεί όνομα χρήστη και κωδικό πρόσβασης MySQL για να μπορούμε να συνδεθούμε στη MySQL. Η σύνδεση γίνεται με την παρακάτω εντολή : mysql -u <username> -p
Η επιλογή -u ακολουθείτε από το username του χρήστη που έχει δημιουργηθεί. Αν δεν έχουμε δημιουργήσει κατάλληλο χρήστη θα πρέπει να συνδεθούμε ως root και να δημιουργήσουμε έναν χρήστη με τα προνόμια που θέλουμε. Δεν είναι καλή τεχνική να χρησιμοποιούμε τον root σαν χρήστη.

Τον χρησιμοποιούμε μόνο όταν θέλουμε να κάνουμε κάποια ενέργεια που απαιτεί δικαίωμα διαχειριστή.
Η επιλογή -p παρακινεί τη Mysql να μας ζητήσει password. Μπορούμε να εισάγουμε τον κωδικό πρόσβασης αμέσως μετά την επιλογή -p χωρίς κενό(π.χ. -p123456). Δεν είναι όμως καλή τεχνική γιατί ο κωδικός είναι εκτεθειμένος. Είναι καλύτερο να μην τον εισάγουμε μαζί με την εντολή ώστε να μας ζητηθεί αμέσως μετά. Έτσι όταν εισάγουμε τον κωδικό δεν θα εμφανίζεται τι πληκτρολογούμε στην οθόνη[4].
Μπορούμε να συνδεθούμε σε συγκεκριμένο host με την επιλογή -h. Αν δεν προσδιορίσουμε τον host που θα συνδεθούμε, σε λογισμικό Unix, ο client επιχειρεί να συνδεθεί στον localhost με Unix socket.[5]
Η επιλογή --protocol δίνει τη δυνατότητα να χρησιμοποιήσουμε ένα συγκεκριμένο πρωτόκολλο μεταφοράς ακόμα και όταν άλλες επιλογές συνήθως έχουν ως αποτέλεσμα τη χρήση διαφορετικού πρωτοκόλλου. Δηλαδή, το --protocol καθορίζει ρητά το πρωτόκολλο μεταφοράς και παρακάμπτει τους προηγούμενους κανόνες, ακόμη και για τον localhost.[5]
Χρησιμοποιούνται ή ελέγχονται μόνο οι επιλογές σύνδεσης που σχετίζονται με το επιλεγμένο πρωτόκολλο μεταφοράς. Άλλες επιλογές σύνδεσης αγνοούνται. Για παράδειγμα, με --host=localhost στο Unix, ο client επιχειρεί να συνδεθεί στον τοπικό server χρησιμοποιώντας ένα Unix socket file, ακόμα κι αν δοθεί μια επιλογή --port ή -P για να καθορίσει TCP/IP port number.
Παράδειγμα απομακρυσμένης σύνδεσης χρησιμοποιώντας την προκαθορισμένη πόρτα 3306 :

```
mysql --host=remote.example.com
```

Υπάρχει δυνατότητα να καθαρίσουμε τις παραμέτρους σύνδεσης χωρίς να χρειάζεται να τις εισάγουμε κάθε φορά, στη γραμμή εντολών, κάθε φορά που θέλουμε να συνδεθούμε.
Τα περισσότερα MySQL προγράμματα, κατά την εκκίνηση, διαβάζουν κάποια αρχεία για να διαμορφώσουν τις επιλογές ρυθμίσεων. Τα αρχεία αυτά ονομάζονται option files ή configuration files. Τα option files παρέχουν έναν βολικό τρόπο για τον καθορισμό επιλογών που χρησιμοποιούνται συχνά, έτσι ώστε να μη χρειάζεται να εισάγονται στη γραμμή εντολών κάθε φορά που εκτελείτε η MySQL.
Τα MySQL προγράμματα διαβάζουν τα option files με την σειρά που εμφανίζονται στον παρακάτω πίνακα

File Name	Purpose
/etc/my.cnf	Global options
/etc/mysql/my.cnf	Global options
SYSCONFDIR/my.cnf	Global options
$MYSQL_HOME/my.cnf	Server-specific options (server only)
defaults-extra-file	The file specified with --defaults-extra-file, if any
~/.my.cnf	User-specific options
~/.mylogin.cnf	User-specific login path options (clients only)
DATADIR/mysqld-auto.cnf	System variables persisted with SET PERSIST or SET PERSIST_ONLY (server only)

Πρέπει να σημειωθεί ότι στις πλατφόρμες UNIX, η MySQL αγνοεί τα option files άμα έχουν write δικαιώματα. Αυτό γίνεται για λόγους ασφαλείας.

Για να μπορέσουμε να συνδεθούμε ως MySQL user χωρίς να εισάγουμε username password στη γραμμή εντολών θα πρέπει να δημιουργήσουμε ένα option file, με όνομα .my.cnf ,να το τοποθετήσουμε στο σωστό path, το οποίο έχει την μορφή :

```
[client]
host=host_name
user=user_name
password=password
```

1.7 Pluggable Authentication

Όταν ο client συνδέεται στον Mysql server, ο server χρησιμοποιεί το username, που του δόθηκε κατά την διάρκεια της σύνδεσης, για να βρει την κατάλληλη γραμμή, που αντιστοιχεί σε αυτόν τον χρήστη, από το table mysql.user. Στη συνέχεια, ο server ελέγχει την ταυτότητα του client, προσδιορίζοντας από τη γραμμή, ποιος έλεγχος ταυτότητας ισχύει για τον client.
Η MySQL περιλαμβάνει πολλά plugins που εφαρμόζουν χαρακτηριστικά ασφαλείας όπως για τον έλεγχο ταυτότητας από Clients που προσπαθούν να συνδεθούν στον MySQL Server. Διατίθενται plugIns για πολλά πρωτόκολλα ελέγχου ταυτότητας. Το plugin που ταιριάζει στην δική μας περίπτωση είναι το Socket Peer-Credential Pluggable Authentication που ελέγχει την ταυτότητα των client που συνδέονται από τον local host μέσω του Unix socket file.[6]
Το server-side auth_socket authentication plugin αυθεντικοποιεί τους χρήστες που συνδέονται από τον localhost μέσω του Unix socket file. Το plugin χρησιμοποιεί το SO_PEERCRED socket option για να πάρει πληροφορίες για τον χρήστη. Έτσι, το plugin αυτό μπορεί να χρησιμοποιηθεί μόνο από συστήματα που υποστηρίζουν το SO_PEERCRED option, όπως τα Linux.[7].
Για να γίνει η εγκατάσταση θα πρέπει να τρέξουμε, στην γραμμή εντολών την εντολή :
INSTALL PLUGIN auth_socket SONAME 'auth_socket.so';
Μπορούμε ελέγξουμε την επιτυχία της εγκατάστασης στον πίνακα INFORMATION_SCHEMA.PLUGINS ή να τρέξουμε την εντολή SHOW PLUGINS.[7]

```
mysql> SELECT PLUGIN_NAME, PLUGIN_STATUS
       FROM INFORMATION_SCHEMA.PLUGINS
       WHERE PLUGIN_NAME LIKE '%socket%';
+-------------+---------------+
| PLUGIN_NAME | PLUGIN_STATUS |
+-------------+---------------+
| auth_socket | ACTIVE        |
+-------------+---------------+
```

Το socket plugin ελέγχει αν ο χρήστης ,που έχει κάνει σύνδεση στο λειτουργικό ταιριάζει με τον χρήστη MYSQL που έχει καθοριστεί να γίνει η ταυτοποίηση με αυτό τον τρόπο.
Στην εντολή δημιουργίας του χρήστη θα πρέπει να καθορίσουμε ότι θα γίνετε η ταυτοποίηση με την χρήση του συγκεκριμένου Plugin :

```
CREATE USER 'user'@'localhost' IDENTIFIED WITH
auth_socket;
```

Σε περίπτωση που υπάρχει ο χρήστης θα πρέπει να εκτελέσουμε [7][7] :

```
ALTER USER 'user'@'localhost' IDENTIFIED WITH auth_socket
```

1.8 Επίλογος

Από όσα εκθέσαμε παραπάνω, εξάγεται το συμπέρασμα ότι στη MySQL και αντίστοιχα στη MariaDB, μπορούμε να συνδεθούμε με πολλούς διαφορετικούς τρόπους. Είναι φανερό ότι , στην περίπτωση αυτής της πτυχιακής, είναι απαραίτητη η χρήση pluggable Authentication για να μπορέσουμε να συνδεθούμε άμεσα σε κάθε server χωρίς να μας ζητηθεί να εισάγουμε username και password.

Κεφάλαιο 2ο: Backup Server

2.1 Αντίγραφο ασφαλείας

Είναι σημαντικό να δημιουργούμε αντίγραφα ασφαλείας των βάσεων δεδομένων, ώστε να μπορούμε να ανακτήσουμε τα δεδομένα σε περίπτωση που παρουσιαστούν προβλήματα, όπως σφάλματα συστήματος, καταστροφή υλικού ή κατά λάθος διαγραφή δεδομένων από χρήστες. Τα αντίγραφα ασφαλείας είναι επίσης απαραίτητα ως προστασία πριν από την αναβάθμιση μιας εγκατάστασης MySQL και μπορούν να χρησιμοποιηθούν για τη μεταφορά μιας εγκατάστασης MySQL σε άλλο σύστημα ή για τη ρύθμιση διακομιστών αντιγράφων.

2.2 Backup types

2.2.1 Physical Vs Logical Backups

Τα φυσικά αντίγραφα ασφαλείας αποτελούνται από αντίγραφα των καταλόγων και των αρχείων που αποθηκεύουν τα περιεχόμενα της βάσης δεδομένων. Αυτός ο τύπος αντιγράφων ασφαλείας είναι κατάλληλος για μεγάλες, σημαντικές βάσεις δεδομένων που πρέπει να ανακτηθούν γρήγορα όταν παρουσιαστούν προβλήματα. Είναι πιο γρήγορος τρόπος από τα λογικά, επειδή περιλαμβάνουν μόνο αντιγραφή αρχείων χωρίς μετατροπή. Για τον λόγο αυτό, το αντίγραφο ασφάλειας που δημιουργείται μπορεί να μεταφερθεί μόνο σε τερματικά που έχουν παρόμοιο Hardware. Αξίζει να σημειωθεί ότι τα φυσικά αντίγραφα ασφαλείας μπορούν να γίνουν μόνο όταν ο MySql server δεν είναι ενεργός. Αν ο server είναι ενεργός θα πρέπει να κλειδωθεί ώστε να μην γίνει κάποια αλλαγή κατά τη διάρκεια της διαδικασίας αντιγράφου ασφάλειας.[8]
Τα λογικά αντίγραφα ασφαλείας αποθηκεύουν πληροφορίες που αντιπροσωπεύονται ως λογική δομή βάσης δεδομένων (CREATE DATABASE, CREATE TABLE) και (INSERT). Αυτός ο τύπος αντιγράφων ασφαλείας είναι κατάλληλος για μικρότερο όγκο δεδομένων όπου μπορούμε να επεξεργαστούμε τις τιμές δεδομένων ή τη δομή του πίνακα ή να δημιουργήσουμε εκ νέου τα δεδομένα σε διαφορετική αρχιτεκτονική. Αυτή δημιουργία αντιγράφων ασφαλείας είναι πιο αργή από τις φυσικές μεθόδους, επειδή ο διακομιστής πρέπει να έχει πρόσβαση στις πληροφορίες της βάσης δεδομένων και να τις μετατρέψει σε λογική μορφή. Τα λογικά αντίγραφα ασφαλείας είναι ανεξάρτητα από τη μηχανή και είναι εξαιρετικά φορητά. Γίνονται κατά την διάρκεια που ο server είναι σε λειτουργία.[8]

2.2.2 Online vs Offline Backups

Τα αντίγραφα ασφαλείας σε σύνδεση πραγματοποιούνται ενώ εκτελείται ο διακομιστής MySQL, έτσι ώστε οι πληροφορίες της βάσης δεδομένων να μπορούν να ληφθούν από τον διακομιστή. Με αυτό τον τρόπο Το αντίγραφο ασφαλείας είναι λιγότερο ενοχλητικό σε άλλους client, οι οποίοι μπορούν να συνδεθούν στον server MySQL κατά τη διάρκεια της δημιουργίας αντιγράφων ασφαλείας και ενδέχεται να έχουν πρόσβαση σε δεδομένα ανάλογα με τις λειτουργίες που πρέπει να εκτελέσουν.
Τα αντίγραφα ασφαλείας εκτός σύνδεσης πραγματοποιούνται ενώ ο διακομιστής είναι σταματημένος. Οι clients μπορεί να επηρεαστούν αρνητικά επειδή ο server δεν είναι διαθέσιμος κατά τη δημιουργία αντιγράφων ασφαλείας. Για το λόγο αυτό, τέτοια αντίγραφα ασφαλείας λαμβάνονται συχνά από ένα αντίγραφο που μπορεί να αφαιρεθεί εκτός σύνδεσης χωρίς να βλάψει τη διαθεσιμότητα.[8]

2.2.3 Τοπικά vs Απομακρυσμένα backups

Ένα τοπικό αντίγραφο ασφαλείας εκτελείται στον ίδιο κεντρικό υπολογιστή όπου εκτελείται ο server MySQL, ενώ ένα απομακρυσμένο αντίγραφο ασφαλείας εκτελείται από διαφορετικό κεντρικό υπολογιστή. Για ορισμένους τύπους αντιγράφων ασφαλείας, το αντίγραφο ασφαλείας μπορεί να ξεκινήσει από έναν απομακρυσμένο κεντρικό υπολογιστή, ακόμη κι αν η έξοδος είναι καθορισμένη τοπικά στον διακομιστή.[8]

2.2.4 Snapshot backups

Ορισμένες υλοποιήσεις συστήματος αρχείων επιτρέπουν τη λήψη «στιγμιότυπων». Αυτά παρέχουν λογικά αντίγραφα του συστήματος αρχείων σε μια δεδομένη χρονική στιγμή, χωρίς να απαιτείται φυσικό αντίγραφο ολόκληρου του συστήματος αρχείων. Η ίδια η MySQL δεν παρέχει τη δυνατότητα λήψης στιγμιότυπων συστήματος αρχείων. Διατίθεται μέσω τρίτων προγραμμάτων όπως Veritas, LVM ή ZFS.[8]

2.2.5 Backup Schedulling

Ο προγραμματισμός δημιουργίας αντιγράφων ασφαλείας είναι πολύτιμος για την αυτοματοποίηση των διαδικασιών δημιουργίας αντιγράφων ασφαλείας. Η ίδια η MySQL δεν παρέχει αυτές τις δυνατότητες.[8]

2.3 Εργαλεία αντιγράφου ασφαλείας

Η MySQL παρέχει κάποια εργαλεία τα οποία μας επιτρέπουν να δημιουργήσουμε αντίγραφο ασφαλείας. Υπάρχουν, όμως, και εξωτερικά εργαλεία τα οποία έχουν καλύτερη απόδοση και μπορούν να ευκολύνουν τον χειριστή DBMS παρέχοντας τον περισσότερες ενέργειες απ’ ότι τα προκαθορισμένα εργαλεία.

2.3.1 Mysqldump

Ένα από τα προ υπάρχοντα εργαλεία αντιγράφου ασφαλείας είναι το mydump. Το πρόγραμμα mysqldump εκτελεί λογικά αντίγραφα ασφαλείας, παράγοντας ένα σύνολο εντολών SQL που μπορούν να εκτελεστούν για την αναπαραγωγή των αρχικών ορισμών αντικειμένων βάσης δεδομένων και δεδομένων πίνακα. Μπορεί να εκτελεστεί σε μία ή περισσότερες βάσεις δεδομένων MySQL για δημιουργία αντιγράφων ασφαλείας ή για μεταφορά σε άλλο SQL server. Η εντολή mysqldump μπορεί επίσης να δημιουργήσει έξοδο σε μορφή CSV, άλλο οριοθετημένο κείμενο ή μορφή XML.[9]
Το mysqldump απαιτεί τουλάχιστον το δικαίωμα SELECT για dumped tables, SHOW VIEW για dumped views, TRIGGER για dumped triggers, LOCK TABLES εάν δεν χρησιμοποιείται η επιλογή --single-transaction και (από MySQL 8.0.21) PROCESS εάν το -- Η επιλογή no-tablespace δεν χρησιμοποιείται. Ορισμένες επιλογές ενδέχεται να απαιτούν άλλα προνόμια όπως αναφέρονται στις περιγραφές των επιλογών.[9]
Τα πλεονεκτήματα του mysqldump περιλαμβάνουν την ευκολία και την ευελιξία της προβολής ή ακόμη και της επεξεργασίας της εξόδου πριν από την επαναφορά. Παρέχει την δυνατότητα κλωνοποίησης βάσεις δεδομένων για ανάπτυξη και εργασίες DBA ή δυνατότητα παραγωγής μικρών παραλλαγών μιας υπάρχουσας βάσης δεδομένων για δοκιμή. Δεν προορίζεται ως γρήγορη ή επεκτάσιμη λύση για τη δημιουργία αντιγράφων ασφαλείας σημαντικών ποσοτήτων δεδομένων. Με

μεγάλα μεγέθη δεδομένων, ακόμα κι αν το βήμα δημιουργίας αντιγράφων ασφαλείας διαρκεί εύλογο χρόνο, η επαναφορά των δεδομένων μπορεί να είναι πολύ αργή, επειδή η επανάληψη των δηλώσεων SQL περιλαμβάνει είσοδο/έξοδο δίσκου για εισαγωγή, δημιουργία και ούτω καθεξής.[9]
Υπάρχουν γενικά τρεις τρόποι χρήσης του mysqldump—προκειμένου να παραχθεί αντίγραφο ασφαλείας από ένα σύνολο ενός ή περισσότερων πινάκων, ένα σύνολο από μία ή περισσότερες πλήρεις βάσεις δεδομένων ή έναν ολόκληρο MySQL server.[9]

```
mysqldump [options] db_name [tbl_name ...]
mysqldump [options] --databases db_name ...
mysqldump [options] --all-databases
```

2.3.2 MyDumper

Το MyDumper είναι ένα εργαλείο το οποίο παράγει λογικά αντίγραφα ασφαλείας για την MySQL. Διαθέτει δύο εργαλεία. Το mydumper το οποίο είναι υπεύθυνο για την εξαγωγή ενός συνεπούς αντιγράφου ασφαλείας. Και το myloader, το οποίο διαβάζει το αντίγραφο ασφαλείας από το mydumper και συνδέεται στην βάση δεδομένων και εισάγει το αντίγραφο ασφαλείας.[10]
Το MyDumper είναι ανοιχτού κώδικα και διατηρείται από την κοινότητα, δεν είναι προϊόν Percona, MariaDB ή MySQL.[10]
Ένα από τα ιδιαίτερα χαρακτηριστικά του mydumper είναι οι multithreading δυνατότητες. Αυτό έχει ως αποτέλεσμα την παράλληλη εκτέλεση και επομένως ταχύτητα και καλύτερη απόδοση. Επίσης παρέχει καλύτερο χειρισμό αντιγράφου ασφαλείας επειδή δημιουργεί ξεχωριστούς φακέλους για τα αρχεία
Για την εκτέλεση του mydumper χρησιμοποιούμε την παρακάτω εντολή η οποία δημιουργεί αντίγραφα για όλες τις βάσεις :

```
Shell> mydumper -uroot –password=<password> --outputdir /backups
```

Αμέσως μετά την εκτέλεση της εντολή τα αρχεία αποθηκεύονται στο φάκελο που καθορίσαμε στην εντολή /backups. Για κάθε βάση υπάρχει το CREATE DATABASE statement στα αρχεία <database_name>-schema-create.sql. Ο κάθε πίνακας (Table) έχει το δικό του schema στα αρχεία <database_name>.<table>-schema.sql και τα αρχεία του κάθε πίνακα βρίσκονται στο <database_name>.<table>.sql. Οι ρουτίνες οι trigger και τα events αποθηκεύονται στα αρχεία <database_name>-schema-post.sql

2.4 Επίλογος

Το αντίγραφο ασφαλείας είναι ένα απολύτως σημαντικό μέρος μιας ολοκληρωμένης στρατηγικής ανάκτησης της βάσης δεδομένων σε περίπτωση καταστροφής.
Σε ένα σύστημα DBMS όπως η MySQL ή MariaDB υπάρχουν διάφοροι τύποι αντιγράφου ασφαλείας που ανάλογα με την περίσταση μπορούμε να τους εφαρμόσουμε.
Υπάρχουν πολλά εργαλεία που μπορούν να εκτελέσουν την διαδικασία αντιγράφου ασφαλείας όπως το mydump που είναι ένα εργαλείο της MySQL και το mydumper το οποίο είναι open source και αναπτύσσεται από την κοινότητα.

Κεφάλαιο 3ο: Monitor Server

3.1 Εισαγωγή

Όπως είχαμε αναφέρει στο προηγούμενο κεφάλαιο, είναι σημαντικό να κρατάμε αντίγραφο ασφαλείας σε περίπτωση που υπάρξει ανάγκη να ανακτήσουμε τα δεδομένα. Πώς όμως μπορούμε να αναγνωρίσουμε πως κάτι δεν λειτουργεί σωστά και να αποτρέψουμε κάποια καταστροφή. Η βάση δεδομένων σχηματίζει ένα κρίσιμο επίπεδο στη στοίβα εφαρμογών. Όλα στην εφαρμογή ή στον ιστότοπο που είναι χτισμένα πάνω σε αυτό το επίπεδο θα εξαρτηθούν από το πόσο καλά αποδίδει η βάση δεδομένων.
Η παρακολούθηση της απόδοσης της βάσης δεδομένων βοηθά στην πρόληψη πιθανών προβλημάτων προτού επηρεάσουν τους τελικούς χρήστες.

3.2 Τι είναι το Monitor

Η παρακολούθηση είναι η τακτική παρατήρηση και καταγραφή των δραστηριοτήτων που λαμβάνουν χώρα σε ένα έργο ή πρόγραμμα. Είναι μια διαδικασία τακτικής συλλογής πληροφοριών για όλες τις πτυχές του έργου. Παρακολούθηση σημαίνει συστηματική και σκόπιμη παρατήρηση και έλεγχος της εξέλιξης των δραστηριοτήτων του προγράμματος.
Η παρακολούθηση περιλαμβάνει επίσης την παροχή πληροφορίας σχετικά με την πρόοδο του προγράμματος.
Αυτή η αναφορά επιτρέπει τη χρήση των συλλεγόμενων πληροφοριών για τη λήψη αποφάσεων για τη βελτίωση της απόδοσης του προγράμματος.[11]

3.3 Μετρήσεις MySQL

Για να παρακολουθήσουμε έναν MySQL server θα πρέπει να μπορούμε να ανακτήσουμε μετρήσεις οι οποίες αφορούν την απόδοση, το χρόνο εκτέλεσης, τον αριθμό των ερωτημάτων που εκτελούνται και την χρήση πόρων. Αυτές οι μετρήσεις μπορούν να αναζητηθούν από τις μεταβλητές κατάστασης server, performance schema, sys schema.

3.3.1 Μεταβλητές κατάστασης Server

Ο MySQL server διατηρεί πολλές μεταβλητές κατάστασης που παρέχουν πληροφορίες σχετικά με τη λειτουργία του. Μπούμε να δούμε αυτές τις μεταβλητές μαζί με τις τιμές τους χρησιμοποιώντας το ερώτημα SHOW GLOBAL STATUS; Η προαιρετική λέξη GLOBAL συγκεντρώνει τις τιμές σε όλες τις συνδέσεις και το SESSION εμφανίζει τις τιμές για την τρέχουσα σύνδεση.[12]
Η οθόνη εμφανίζει την παρακάτω μορφή :

```
mysql> SHOW GLOBAL STATUS;
+-----------------------------------+------------+
| Variable_name                     | Value      |
+-----------------------------------+------------+
| Aborted_clients                   | 0          |
| Aborted_connects                  | 0          |
| Bytes_received                    | 155372598  |
| Bytes_sent                        | 1176560426 |
...
| Connections                       | 30023      |
| Created_tmp_disk_tables           | 0          |
```

```
| Created_tmp_files                 | 3          |
| Created_tmp_tables                | 2          |
...
| Threads_created                   | 217        |
| Threads_running                   | 88         |
| Uptime                            | 1389872    |
+-----------------------------------+------------+
```

3.3.2 Performance Schema

Το MySQL Performance Schema είναι μια δυνατότητα παρακολούθησης της εκτέλεσης του MySQL Server σε χαμηλό επίπεδο. Το Performance Schema είναι διαθέσιμο ως performance_schema Database που περιέχει πίνακες με πληροφορίες σχετικά με την επίδοση του server και μπορούμε να τις ανακτήσουμε με SELECT ερωτήματα.[13]
Για να σιγουρέψουμε ότι το Performance Schema είναι διαθέσιμο χρησιμοποιούμε το ερώτημα

```
mysql> SHOW VARIABLES LIKE 'performance_schema';
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| performance_schema | ON    |
+--------------------+-------+
```

Για να δούμε όλους τους πίνακες του Performance Schema εκτελούμε το ερώτημα[14]

```
mysql> SHOW TABLES FROM performance_schema;
+----------------------------------------------------+
| Tables_in_performance_schema                       |
+----------------------------------------------------+
| accounts                                           |
| cond_instances                                     |
| events_stages_current                              |
| events_stages_history                              |
| events_stages_history_long                         |
...
```

Το παρακάτω παράδειγμα μας δίνει ποιο ερώτημα έχει τον περισσότερο χρόνο εκτέλεσης [14]

```
mysql> SELECT digest_text, avg_timer_wait
       FROM performance_schema.events_statements_summary_by_digest
       ORDER BY avg_timer_wait DESC
       LIMIT 1
+--------------------------------------------+----------------+
| digest_text                                | avg_timer_wait |
+--------------------------------------------+----------------+
| INSERT INTO `rental` VALUES (...) /* , ... */ |   407201600000 |
+--------------------------------------------+----------------+
1 row in set (0.0052 sec)
```

3.3.3 Sys Schema

Επειδή το Performance Schema είναι χαμηλού επιπέδου είναι δύσκολο στον χειρισμό. Από την MySQL 5.7 και έπειτα περιλαμβάνεται το sys Schema, μια σειρά αντικειμένων τα οποία συνοψίζουν τα δεδομένα που συλλέγονται από το Performance Schema σε μια κατανοητή μορφή.[15]
Παράδειγμα που συνοψίζει το αρχείο I/O ομαδοποιημένο ανά host

```
mysql> SELECT * FROM sys.host_summary_by_file_io;
+------------+-------+------------+
| host       | ios   | io_latency |
+------------+-------+------------+
| localhost  | 67570 | 5.38 s     |
| background |  3468 | 4.18 s     |
+------------+-------+------------+
```

3.4 Performance Monitoring Tools

Ένας διαχειριστής βάσεων δεδομένων, για να μπορεί να έχει ευκολία στην παρακολούθηση των μετρήσεων είναι απαραίτητο να χρησιμοποιήσει ένα εργαλείο παρακολούθησης(Monitor tool). Τα εργαλεία παρακολούθησης MySQL υποβάλλουν τακτικά αιτήματα για την ανάκτηση στατιστικών από το Performance Schema της MySQL. Κάθε απάντηση περιέχει πληροφορίες για το σύστημα MySQL. Αυτές οι πληροφορίες στη συνέχεια μεταφράζονται σε μετρήσεις και παρουσιάζονται σε γραφικό περιβάλλον και παρέχουν ειδοποίηση όταν ξεφεύγουν από την κανονική τους μορφή. Τέτοια εργαλεία είναι MySQL Enterprise Monitor , MySQL Workbench και πολλά άλλα.

3.4.1 MySQL Workbench

Το MySQL Workbench είναι ένα εργαλείο μοντελοποίησης για βάσεις δεδομένων MySQL, το οποίο εκτός από τις πολλές χρήσιμες και προηγμένες λειτουργίες του, παρέχει ένα σύνολο εργαλείων για την προβολή και τη βελτίωση της απόδοσης της βάσης δεδομένων. Μπορεί να απεικονίσει τις κύριες μετρήσεις απόδοσης, όπως εισερχόμενη και εξερχόμενη κίνηση δικτύου, στατιστικά στοιχεία απόδοσης, εκτελεσμένες δηλώσεις SQL, κατάσταση InnoDB, συμπεριλαμβανομένης της δραστηριότητας δίσκου, εγγραφής και ανάγνωσης. Επιπλέον, με τις αναφορές απόδοσης, είναι πολύ πιο εύκολο να αναλυθεί η απόδοση της βάσης δεδομένων MySQL.[16]

3.5 Επίλογος

Γίνεται , λοιπό, αντιληπτό ότι η παρακολούθηση ενός server είναι απαραίτητη . Ο τρόπος με τον οποίο μπορεί να γίνει η παρακολούθηση αυτή ,συγκεκριμένα σε ένα DBMS MySQL/MariaDB, είναι είτε χαμηλού επιπέδου εκτελώντας ερωτήματα (query) στο performance schema είτε υψηλού επιπέδου με εργαλεία παρακολούθησης, όπως το MySQL Workbench.

Κεφάλαιο 4ο: Backup Servers Η λύση στο πρόβλημα

4.1 Εισαγωγή

Για να ικανοποιήσουμε το ζητούμενο της εργασίας το οποίο είναι η ημιαυτοματοποιημένη διαδικασία αντιγράφου ασφαλείας σε διάφορους server που βρίσκονται σε διαφορετικά τερματικά θα πρέπει να αναπτύξουμε μια διαδικασία ρουτίνας. Η διαδικασία αυτή θα βρίσκεται σε ένα τερματικό που θα έχει ανατεθεί για τον σκοπό αυτό. Η διαδικασία αυτή θα μπορέσει να συνδεθεί, απομακρυσμένα, σε κάθε ένα τερματικό και έπειτα στο MySQL/MariaDB server, θα δημιουργεί ένα λογικό αντίγραφο ασφαλείας και θα το αποθηκεύει στο τερματικό, που εκτελεί την ρουτίνα, με όνομα φακέλου την ημερομηνία, την ώρα και το όνομα του server όπου ανήκει το αντίγραφο.
Η διαδικασία αυτή θα μπορούσε να γίνει με δύο τρόπους. Ο πρώτος είναι να χρησιμοποιήσουμε ένα έτοιμο πρόγραμμα το οποίο μπορεί να συνδέεται αυτόματα σε κάθε ένα τερματικό και να δημιουργεί αντίγραφο ασφαλείας. Όμως τα προγράμματα αυτά είναι με πληρωμή οπότε για ευνόητους λόγους δεν μπορούν να χρησιμοποιηθούν. Ο δεύτερος τρόπος είναι να δημιουργήσουμε εμείς ένα πρόγραμμα.
Ο τρόπος που χρησιμοποιήθηκε για αυτή την εργασία είναι η ανάπτυξη ενός script το οποίο δέχεται ένα αρχείο με τις πληροφορίες που αφορούν τους servers, το διαβάζει γραμμή γραμμή, εξασφαλίζει μία σύνδεση με ssh, εκτελεί το εργαλείο mydumper και αποθηκεύει τον φάκελο με τα αρχεία αντιγράφου ασφαλείας. Τέλος για να μπορεί το script αυτό να εκτελείται σε προγραμματισμένο χρόνο, εφαρμόστηκε η υπηρεσία crontab με την οποία προγραμματίζεται πότε θα εκτελεστεί μια εντολή.

4.2 Shell Script

4.2.1 Τι είναι το shell script

Ένα shell script είναι ένα αρχείο κειμένου το οποίο μπορεί να δημιουργηθεί με έναν επεξεργαστή κειμένου όπως nano , vi. Το αρχείο αυτό περιέχει εντολές για το shell. Το script πρέπει να ξεκινάει με την πρώτη γραμμή που καθορίζει ποιο shell αφορούν οι εντολές που θα ακολουθήσουν. Στην συγκεκριμένη περίπτωση χρησιμοποιούμε το bash shell. Οπότε η πρώτη γραμμή θα είναι :

```
#!/bin/bash
```

Για να μπορέσει ένα script να εκτελεστεί θα πρέπει να έχει άδεια πρόσβασης execute.
Για να εκτελέσουμε ένα script πρέπει να αναφερθούμε στο όνομα αρχείου είτε με σχετική είτε με απόλυτη διαδρομή.

4.2.2 Παράμετρος στο script

Κάθε φορά που το κέλυφος διερμηνεύει μια εντολή συνδέει ονόματα μεταβλητών σε κάθε παράμετρο της γραμμής εντολών. Ως παράμετροι της γραμμής εντολών θεωρούνται ακολουθίες χαρακτήρων διαχωριζόμενες με κενά ή tab. Μια παράμετρός μπορεί να είναι μία μεταβλητή μια εντολή ή ένα αρχείο.
bash scriptname.sh parameter1 parameter2 parameter3
Η $0 αφορά το όνομα του script.
Για να χρησιμοποιήσουμε τις παραμέτρους στο script χρησιμοποιούμε τις ειδικές μεταβλητές $1-$9

$1 αφορά την πρώτη παράμετρο το $2 την δεύτερη και ούτω κάθε εξής.
Για να διαβάσουμε το πλήθος των παραμέτρων χρησιμοποιούμε την $#. Για να διαβάσουμε όλες τις μεταβλητές σε ένα string $*.

4.2.3 Η συνθήκη if και η εντολή test

Όπως σε όλα τα προγραμματιστικά περιβάλλοντα έτσι και στο bash script υπάρχουν εντολές συνθήκης όπως η if.

```
if εντολή ελέγχου ;
then
Commands….
elif εντολή ελέγχου;
then
Commands….
else
Commands….
Fi
```

Συνήθως, θέλουμε να ελέγξουμε το αποτέλεσμα μιας αριθμητικής συνθήκης ή μια σύγκριση από strings. Η if από μόνη της δεν είναι ικανή να κάνει αυτούς τους ελέγχους. Υπάρχει η εντολή test, η οποία δέχεται ως όρισμα μια συνθήκη-έλεγχο και επιστέφει ως exit code: true (0) αν η συνθήκη είναι αληθής , false (>0) αν η συνθήκη είναι ψευδής. Η εντολή test μπορεί να πραγματοποιήσει κατηγορίες ελέγχων όπως strings, ακεραίους, αρχεία και να υπολογίσει λογικούς τελεστές(AND,OR,NOT).
Η εντολή test έχει και μία διαφορετική σύνταξη και έχει την μορφή [. Για τους υπόλοιπους ελέγχους που μπορεί να πραγματοποιήσει η εντολή test, δηλαδή η εντολή [, το bash έχει αντίστοιχη εσωτερική εντολή (built-in): την [[ ]]. Η χρήση των [[ ]] είναι παρόμοια με την χρήση των [ ], μόνο που τους ελέγχους τους κάνει το ίδιο το shell και δεν εκτελείται η επιπλέον εντολή (η test). άρα η χρήση των [[ ]] υπερτερεί στην ταχύτητα εκτέλεσης.

4.2.4 Βρόχος while do και ανάγνωση γραμμής αρχείου

Η εντολή while χρησιμοποιείται για την επανειλημμένη εκτέλεση μιας λίστας εντολών. Η σύνταξη είναι :

```
while [ condition ]; do COMMAND; done
```

Είναι χρήσιμο το γεγονός ότι η εντολή while έχει την δυνατότητα να διαβάζει ένα αρχείο κειμένου γραμμή γραμμή. Η αντίστοιχη σύνταξη είναι :

```
while read -r line; do COMMAND; done < input.file
```

Η επιλογή -r αποτρέπει την ερμηνεία backslash escape character.[17]

4.3 Η λίστα με τους servers

Το εκτελέσιμο αρχείο θα πρέπει να δέχεται σαν παράμετρο μια λίστα με τους servers που θα ορίσει ο χρήστης. Το shell script μπορεί να δεχτεί ένα txt αρχείο και να διαβάζει γραμμή γραμμή. Μέσα από τις κατάλληλες εντολές μπορούμε να επεξεργαστούμε, να χωρίσουμε την κάθε γραμμή. Οπότε αν το txt έχει την μορφή γραμμές και στήλες, όπου κάθε γραμμή και ο κάθε server και κάθε στήλη η πληροφορία σύνδεσης, 1 τύπος σύνδεσης 2 ip/uri server 3 χρήστης σύνδεσης 4 private key για σύνδεση(π.χ. S web.iee.ihu.gr webbackup ~/.ssh/id_rsa). Ο τύπος σύνδεσης αναφέρεται στην

μεταβλητή χαρακτήρα η οποία ορίζει τον τρόπο με τον οποίο θα γίνει η σύνδεση SSH. Αν τύπος σύνδεσης “$type” = S τότε η σύνδεση SSH γίνεται μέσο socket file. Αν “$type”=P τότε η σύνδεση SSH γίνεται με port forwarding.

Παράδειγμα αρχείου:

```
S web.iee.ihu.gr ieebackup ~/.ssh/id_rsa
P     aboard.iee.ihu.gr     ieebackup     ~/.ssh/id_rsa_aboard     127.0.0.1
ieebackupaboard
```

4.4 Απομακρυσμένη σύνδεση με SSH

Για να μπορέσουμε να κάνουμε οποιαδήποτε ενέργεια backup θα πρέπει να συνδεθούμε απομακρυσμένα με τα τερματικά που φιλοξενούν τις βάσεις δεδομένων. Ο τρόπος με τον οποίο μπορεί να γίνει μια ασφαλή απομακρυσμένη σύνδεση είναι με το πρωτόκολλο SSH. Το πρωτόκολλο SSH χρησιμοποιεί κρυπτογράφηση για να εξασφαλίσει τη σύνδεση μεταξύ ενός client και ενός server. Όλα τα στοιχεία ελέγχου ταυτότητας χρήστη, εντολές, έξοδοι και μεταφορές αρχείων είναι κρυπτογραφημένα για προστασία από επιθέσεις στο δίκτυο[18].
Ένα ακόμα θετικό προνόμιο που έχει το SSH είναι ο αυτόματος τρόπος σύνδεσης. Το πρωτόκολλο SSH υποστηρίζει πολλές μεθόδους ελέγχου ταυτότητας. Αναμφισβήτητα ένα από τα πιο σημαντικά από αυτά είναι ο έλεγχος ταυτότητας με δημόσιο κλειδί(public key authentication) για δια δραστικές και αυτοματοποιημένες συνδέσεις [18]
Ο έλεγχος ταυτότητας με δημόσιο κλειδί παρέχει κρυπτογραφική ισχύ που ακόμη και οι εξαιρετικά μεγάλοι κωδικοί πρόσβασης δεν μπορούν να προσφέρουν. Με το SSH, ο έλεγχος ταυτότητας με δημόσιο κλειδί βελτιώνει σημαντικά την ασφάλεια καθώς απαλλάσσει τους χρήστες από το να θυμούνται περίπλοκους κωδικούς πρόσβασης (ή ακόμα χειρότερα, να τους καταγράφουν)[19]. Τα κλειδιά SSH μπορούν να χρησιμοποιηθούν για την αυτοματοποίηση της πρόσβασης σε server. Παρέχουν επίσης ενιαία σύνδεση, επιτρέποντας στον χρήστη να μετακινείται μεταξύ των λογαριασμών του/της χωρίς να χρειάζεται να πληκτρολογεί κωδικό πρόσβασης κάθε φορά[18].
Όπως συμβαίνει με κάθε σχήμα κρυπτογράφησης, ο έλεγχος ταυτότητας με δημόσιο κλειδί βασίζεται σε έναν αλγόριθμο. Υπάρχουν αρκετοί καλά ερευνημένοι, ασφαλείς και αξιόπιστοι αλγόριθμοι εκεί έξω - οι πιο συνηθισμένοι είναι οι RSA και DSA. Σε αντίθεση με τους αλγορίθμους κρυπτογράφησης (συμμετρικού ή μυστικού κλειδιού), οι αλγόριθμοι κρυπτογράφησης δημόσιου κλειδιού λειτουργούν με δύο ξεχωριστά κλειδιά. Αυτά τα δύο κλειδιά σχηματίζουν ένα ζεύγος που είναι συγκεκριμένο για κάθε χρήστη[19]
Οποιαδήποτε υλοποίηση SSH παρέχει την δυνατότητα, στους χρήστες, να δημιουργήσουν ένα ζευγάρι κλειδιών. Το ζευγάρι αποτελείται από δύο κλειδιά [19]:

- Ένα δημόσιο κλειδί που υπάρχει στον server SSH. Οποιοσδήποτε έχει αντίγραφο του δημόσιου κλειδιού μπορεί να κρυπτογραφήσει δεδομένα τα οποία στη συνέχεια μπορούν να διαβαστούν μόνο από το άτομο που κατέχει το αντίστοιχο ιδιωτικό κλειδί. Μόλις ένας server SSH λάβει ένα δημόσιο κλειδί από έναν χρήστη και θεωρήσει το κλειδί αξιόπιστο, ο server επισημαίνει το κλειδί ως εξουσιοδοτημένο στο αρχείο authorized_keys. Τέτοια κλειδιά ονομάζονται εξουσιοδοτημένα κλειδιά.
- Ένα ιδιωτικό κλειδί που παραμένει (μόνο) στον χρήστη. Η κατοχή αυτού του κλειδιού αποτελεί απόδειξη της ταυτότητας του χρήστη. Μόνο ένας χρήστης που έχει στην κατοχή του ένα ιδιωτικό κλειδί που αντιστοιχεί στο δημόσιο κλειδί του server θα μπορεί να πραγματοποιήσει επιτυχή έλεγχο ταυτότητας. Τα ιδιωτικά κλειδιά πρέπει να αποθηκεύονται

και να χειρίζονται προσεκτικά και δεν πρέπει να διανέμονται αντίγραφα του ιδιωτικού κλειδιού. Τα ιδιωτικά κλειδιά που χρησιμοποιούνται για τον έλεγχο ταυτότητας χρήστη ονομάζονται κλειδιά ταυτότητας.

Το πρώτο βήμα που πρέπει να κάνουμε είναι να δημιουργήσουμε το ζευγάρι κλειδιών με την εντολή $ ssh-keygen .

Έτσι θα δημιουργηθούν τα RSA κλειδιά 2048 bit το οποίο είναι αρκετά ασφαλή για τις περισσότερες περιπτώσεις. Μπορούμε να προσθέσουμε την επιλογή -b 4096 για να δημιουργήσουμε ένα μεγαλύτερο κλειδί 4096 bit[6]. Ο κατάλληλος φάκελος που φιλοξενεί τα κλειδιά είναι ο .ssh. αν δεν υπάρχει θα πρέπει να τον δημιουργήσουμε. Τα κλειδιά που έχουν δημιουργηθεί έχουν όνομα id_rsa και id_rsa.pub. Θα πρέπει να μεταφέρουμε το public key στον server που θέλουμε να κάνουμε σύνδεση. Το τερματικό που είναι υπεύθυνο για να συνδέεται με τους servers θα έχει τα private keys στον φάκελο ~/.ssh. Με αυτόν τον τρόπο μπορούμε πλέον να εκτελέσουμε την εντολή ssh user@server και να συνδεθούμε χωρίς να μας ζητηθεί κωδικός πρόσβασης το οποίο θα μας επιτρέψει να εκτελέσουμε μία αυτοματοποιημένη διαδικασία (π.χ. shell script) χωρίς διακοπή.

4.5 Σύνδεση στη mysql χωρίς κωδικό

Εφόσον έχουμε συνδεθεί σε έναν server με SSH, για να μπορέσουμε να εκτελέσουμε οποιαδήποτε ενέργεια θα πρέπει να συνδεθούμε στον mysql/mariadb server. Αυτή η διαδικασία δεν μπορεί να γίνει αυτοματοποιημένη γιατί θα πρέπει για κάθε server να εισάγουμε username και password. Η ιδανική περίπτωση θα ήταν με το που εξασφαλιστεί η SSH σύνδεση να μπορούμε να συνδεθούμε στον MySQL/MariaDB server χωρίς να εισάγουμε username και password.

Σε προηγούμενο κεφάλαιο είχαμε εξηγήσει πως ένας από τους τρόπους σύνδεσης σε έναν MySQL/MariaDB server γίνεται με plugin και συγκεκριμένα με το Pluggable Authentication. Στην ουσία αυτό που θα πρέπει να έχουμε προετοιμάσει σε κάθε server είναι να έχουμε δημιουργήσει έναν χρήστη, ο οποίος έχει, προφανώς, δικαιώματα για να μπορεί να πάρει το αντίγραφο ασφαλείας και να μπορεί να κάνει σύνδεση με pluggable authentication. Καλό θα είναι ο χρήστης που θα κάνουμε την SSH σύνδεση να έχει ίδιο username με αυτόν που θα αναθέσουμε την Pluggable Authentication. Το pluggable authentication, στην ουσία, δίνει την δυνατότητα σύνδεσης, χωρίς εισαγωγή username password επειδή κάνει την αυθεντικοποίηση με τα στοιχεία που έχει κάνει σύνδεση ο χρήστης linux. Για παράδειγμα, αν συνδεθούμε με το SSH ως user1, και αυτός ο χρήστης έχει pluggable authentication στην MySQL/MariaDB, θα μπορεί να συνδεθεί στον server χωρίς username password.

4.6 Δημιουργία Shell Script

Μέχρι αυτό το σημείο έχουμε επιλέξει το εργαλείο που θα κάνει backup ,έχουμε βρει τρόπο σύνδεσης στον server και έχουμε δημιουργήσει έναν χρήστη στο DBMS που μπορεί να συνδεθεί, από το linux socket, χωρίς κωδικό. Με την δημιουργία script θα μπορέσουμε να εκτελέσουμε την διαδικασία σύνδεσης και backup. Θα πρέπει να λάβουμε υπόψη ότι το script θα πρέπει να συνδέεται σε κάθε server να εκτελεί τον mydumper χωρίς να υπάρχει διακοπή από κάποια εντολή.

4.6.1 To Script

Το πρώτο βήμα θα πρέπει να κάνουμε έλεγχο, αν υπάρχει και είναι αρχείο, στην πρώτη παράμετρο που δέχεται το script.

```
if [ ! -f $1 ] ; then
       echo "$1 does not exist or is not a file"
       exit 1
```

```
fi
```

Στη συνέχεια θα διαβάσουμε το αρχείο γραμμή γραμμή.

```
while read p; do
…
done <$1
```

Εφόσον διαβάζουμε κάθε γραμμή ξεχωριστά θα πρέπει αρχικά να κάνουμε έλεγχο για το αν αυτή η γραμμή είναι κενή ώστε να την παραλείψουμε και να πάμε στην επόμενη.

```
if [[ -z $p ]] ; then
            continue
fi
```

Μετά από τους παραπάνω ελέγχους μπορούμε να χωρίσουμε την γραμμή στις κατάλληλες μεταβλητές σύμφωνα με την μορφή που είχαμε ορίσει στο αρχείο με την λίστα server.

```
type=$(echo $p | awk '{ print $1 }')
server=$(echo $p | awk '{ print $2 }')
user=$(echo $p | awk '{ print $3 }')
path=$(echo $p | awk '{ print $4 }')
```

Ορίζουμε σε μία μεταβλητή την τωρινή ημερομηνία και συντάσσουμε το όνομα του φακέλου των αρχείων αντιγράφου ασφαλείας.

Με αυτόν τον τρόπο θα αναγνωρίσουμε τον φάκελο του backup του κάθε server με την ώρα που έχει ολοκληρωθεί η διαδικασία.

Στη συνέχεια κάνουμε έλεγχο την μεταβλητή type για να αναγνωρίσουμε την διαδικασία που θα πρέπει να ακολουθήσουμε για να κάνουμε σύνδεση με SSH.

```
if [[ "$type" == S ]] ; then
```

Αν στον server μπορούμε να συνδεθούμε με socket file τότε θα δημιουργήσουμε το socket file το οποίο κοιτάει στο file της mysql. Επειδή με την εντολή ssh θα συνδεθούμε στο λειτουργικό περιβάλλον του server θα πρέπει να εκτελέσουμε την εντολή αυτή στο background

```
ssh -L ~/sock:/run/mysqld/mysqld.sock -i $path $user@$server -n "sleep 10000" &
```

Πριν κάνουμε οποιαδήποτε άλλη ενέργεια θα πρέπει να κρατήσουμε το process id για να μπορέσουμε να χειριστούμε την εντολή αργότερα.

pid=$!

Είναι απαραίτητο να εκτελέσουμε την εντολή sleep για να υπάρχει μια συνεχής ροή στο script και να μπορέσουν οι εντολές να εκτελεστούν με την σειρά που πρέπει.

sleep 5

Σε αυτό το σημείο εκτελούμε την εντολή mydumper δίνοντας την επιλογή socket file, τον χρήστη mysql που έχει δικαίωμα backup και το όνομα του φακέλου των αρχείων αντιγράφου ασφάλειας.

```
mydumper -S ~/sock -u $user -o $file
sleep 1
```

Στο τέλος θα πρέπει να κλείσουμε το socket file και να σταματήσουμε την σύνδεση ssh που τρέχει στο παρασκήνιο.

```
rm ~/sock
kill $pid
```

Στην περίπτωση που μπορούμε να συνδεθούμε με port forwarding η διαδικασία είναι η ίδια με την socket file διαφορά ότι το ssh γίνεται με port forwarding και η εντολή mydumper έχει την επιλογή -P port.
Αρχικά πρέπει να διακρίνουμε τα έξτρα πεδία που υπάρχουν στην γραμμή, από το αρχείο servers, που μας παρέχουν την πληροφορία για το port forwarding.

```
hostname=$(echo $p | awk '{ print $5 }')
password=$(echo $p | awk '{ print $6 }')
```

Η διαδικασία που ακολουθεί είναι παρόμοια με την socket file απλά οι εντολές έχουν τα option για το port forwarding.

```
ssh -L 3336:$hostname:3306 $user@$server -i $path -n sleep 10000 &
pid=$!
sleep 5
mydumper -P 3336 -u $user -p $password -h $hostname -o $file
sleep 1
kill $pid
```

Σε αυτό το σημείο μπορούμε να εμφανίσουμε ότι η διαδικασία έχει ολοκληρωθεί.

```
echo "Backup to $p OK"
```

Ολοκληρωμένο το script.

```
#/bin/bash
if [ ! -f $1 ] ; then
      echo "$1 does not exist or is not a file"
      exit 1
fi

while read p; do
      if [[ -z $p ]] ; then
            continue
      fi
      type=$(echo $p | awk '{ print $1 }')
      server=$(echo $p | awk '{ print $2 }')
      user=$(echo $p | awk '{ print $3 }')
      path=$(echo $p | awk '{ print $4 }')
      d=`date +'%Y%m%d_%H%M%S'`
      file=$server"_"$d
      sleep 2
      if [[ "$type" == S ]] ; then
                  ssh -L ~/sock:/run/mysqld/mysqld.sock -i $path $user@$server
-n "sleep 10000" &
            pid=$!
            sleep 5
            mydumper -S ~/sock -u $user -o $file
            sleep 1
            rm ~/sock
```

```
            kill $pid
      else
            hostname=$(echo $p | awk '{ print $5 }')
            password=$(echo $p | awk '{ print $6 }')
            ssh -L 3336:$hostname:3306 $user@$server -i $path -n sleep 10000 &
            pid=$!
            sleep 5
            mydumper -P 3336 -u $user -p $password -h $hostname -o $file
            sleep 1
            kill $pid
      fi
      echo "Backup to $p OK"
done <$1
```

4.7 CronTab και προγραμματισμένη εκτέλεση script

Έχουμε ολοκληρώσει την διαδικασία backup servers και το μόνο που έχει μείνει είναι να προγραμματίσουμε το πότε θα εκτελείται. Το λειτουργικό linux έχει μια υπηρεσία που ταιριάζει απόλυτα στο ζητούμενο.

Η υπηρεσία cron επιτρέπει στους sysadmins να προγραμματίζουν εργασίες που θα εκτελούνται σε μια συγκεκριμένη στιγμή στο μέλλον. Η υπηρεσία cron μπορεί να προγραμματίζει εργασίες σε επαναλαμβανόμενη βάση, όπως ημερήσια, εβδομαδιαία ή μηνιαία[20].

Το πρόγραμμα cron εκτελείται με βάση εντολές που καθορίζονται σε έναν πίνακα cron (crontab). Κάθε χρήστης, συμπεριλαμβανομένου του root, μπορεί να έχει ένα αρχείο cron. Αυτά τα αρχεία δεν υπάρχουν από προεπιλογή, αλλά μπορούν να δημιουργηθούν στον κατάλογο /var/spool/cron χρησιμοποιώντας την εντολή crontab -e που χρησιμοποιείται επίσης για την επεξεργασία ενός αρχείου cron[20]. Μέσα στο αρχείο υπάρχουν αναλυτικά σχόλια που εξηγούν την χρήση του crontab.

Η εντολή έχει την μορφή : m h dom mon dow command

m=λεπτό της ώρας(0-59), h=ώρα της ημέρας(0-23), dom=μέρα του μήνα(1-31), mon=μήνας(1-12), dow=ημέρα της εβδομάδος(0-6).

Παράδειγμα, θέλουμε να εκτελέσουμε backup στους servers την Δευτέρα στις 5 π.μ. κάθε εβδομάδα:

0 5 * * 1 bash ~/script.sh servers

4.8 Επίλογος

Είναι σημαντικό για οποιοδήποτε dbms να μπορεί να δημιουργεί αντίγραφο ασφαλείας. Όμως όταν υπάρχουν πολλαπλά συστήματα το κάθε ένα σε διαφορετικό τερματικό ο προκαθορισμένος τρόπος είναι χρονοβόρος ή ακόμα και αδύνατος σε περίπτωση υπερβολικού αριθμού server. Για να βρούμε λύση σε αυτό το πρόβλημα καταφέραμε και δημιουργήσαμε ένα εκτελέσιμο αρχείο (shell script) το οποίο δέχεται μία λίστα με τα στοιχεία σύνδεσης του κάθε server, μπορεί και συνδέεται με SSH, είτε με socket file είτε με port forwarding, εκτελεί την εντολή mydumper, όπου έχει καλύτερη απόδοση από το προϋπάρχουν εργαλείο της mysql το mydump. Στο τέλος, αν θέλουμε να προγραμματίσουμε την ώρα και την στιγμή όπου θα εκτελείται όλη αυτή η διαδικασία χρησιμοποιούμε την υπηρεσία, που υπάρχει στα λειτουργικά συστήματα linux, το crontab. Έτσι μπορούμε να κάνουμε μία αυτοματοποιημένη διαδικασία αντιγράφου ασφαλείας όπου το μόνο που χρειάζεται είναι μια λίστα με τα στοιχεία των server και τον προγραμματισμό της υπηρεσίας crontab ώστε να εκτελεί το script σε

Κεφάλαιο 5ο: Παρακολούθηση Συστημάτων Η λύση στο πρόβλημα

5.1 Εισαγωγή

Είναι πολύ σημαντικό σε οποιοδήποτε υπολογιστικό σύστημα να υπάρχει μια μορφή παρακολούθησης. Ο χειριστής πρέπει να γνωρίζει την κατάσταση του συστήματος τόσο του τεχνικού κομματιού, δηλαδή αν υπάρχει φυσική βλάβη, έχει γεμίσει ο αποθηκευτικός χώρος, τόσο και του λειτουργικού, δηλαδή αν

είναι ενεργός ο server, αν ο χρόνος απόκρισης είναι μεγάλος ή μικρός. Στην περίπτωση πολλαπλών συστημάτων είναι αναγκαία η χρήση εργαλείου παρακολούθησης ώστε να διευκολύνει σε μεγάλο βαθμό τον τρόπο και να μικρύνει τον χρόνο που χρειάζεται η διαδικασία παρακολούθησης.

5.2 Zabbix

Το κατάλληλο εργαλείο για την ταυτόχρονη και λεπτομερή παρακολούθηση πολλαπλών server είναι το zabbix.

Το Zabbix είναι ένα λογισμικό που παρακολουθεί πολυάριθμες παραμέτρους ενός δικτύου και την υγεία και την ακεραιότητα server, εικονικών μηχανών, εφαρμογών, υπηρεσιών, βάσεων δεδομένων, τοποθεσιών web, του cloud και πολλά άλλα. Το Zabbix χρησιμοποιεί έναν ευέλικτο μηχανισμό ειδοποίησης που επιτρέπει στους χρήστες να διαμορφώνουν ειδοποιήσεις που βασίζονται σε email για σχεδόν οποιοδήποτε συμβάν. Αυτό επιτρέπει μια γρήγορη αντίδραση σε προβλήματα server. Το Zabbix προσφέρει εξαιρετικές δυνατότητες αναφοράς και οπτικοποίησης δεδομένων με βάση τα αποθηκευμένα δεδομένα. Αυτό καθιστά το Zabbix ιδανικό για προγραμματισμό χωρητικότητας[21].

Το Zabbix υποστηρίζει τόσο polling όσο και trapping. Όλες οι αναφορές και τα στατιστικά στοιχεία του Zabbix, καθώς και οι παράμετροι διαμόρφωσης, είναι προσβάσιμα μέσω μιας διεπαφής web. Ένα web-based frontend διασφαλίζει ότι η κατάσταση του δικτύου και η υγεία των server μπορούν να αξιολογηθούν από οποιαδήποτε τοποθεσία. Με τη σωστή διαμόρφωση, το Zabbix μπορεί να παίξει σημαντικό ρόλο στην παρακολούθηση της υποδομής πληροφορικής. Αυτό ισχύει εξίσου για μικρούς οργανισμούς με λίγους server και για μεγάλες εταιρείες με πλήθος server[21].

Το Zabbix είναι δωρεάν. Το Zabbix είναι γραμμένο και διανέμεται σύμφωνα με την έκδοση 2 της Γενικής Άδειας Δημόσιας Χρήσης GPL. Σημαίνει ότι ο πηγαίος κώδικας του διανέμεται ελεύθερα και είναι διαθέσιμος για το ευρύ κοινό.

5.3 Αρχιτεκτονική Zabbix

Το Zabbix αποτελείτε από πολλά στοιχεία λογισμικού.

Το πρώτο στοιχείο είναι ο Zabbix server ο οποίος είναι το κύριο συστατικό το οποίο συγκεντρώνεται όλη οι πληροφορία. Ο Zabbix server είναι το κεντρικό σημείο όπου αποθηκεύονται όλα τα στατιστικά δεδομένα.

Το Zabbix χρειάζεται μια βάση δεδομένων ώστε να μπορεί να αποθηκεύει τις πληροφορίες και τα δεδομένα που συγκεντρώνει.

Για να μπορεί ο χρήστης να έχει μια εύκολη πρόσβαση στο Zabbix από οπουδήποτε και σε οποιαδήποτε πλατφόρμα, παρέχεται σε αυτόν μία διεπαφή χρήστη. Η διεπαφή είναι κομμάτι του Zabbix server και συνήθως αλλά όχι απαραίτητα, τρέχει στο ίδιο τερματικό με τον server.

Οι Zabbix agents εφαρμόζονται σε τερματικά για να παρακολουθούν ενεργά τους τοπικούς πόρους και τις εφαρμογές και να αναφέρουν τα συγκεντρωμένα δεδομένα στον διακομιστή Zabbix. Από το Zabbix 4.4, υπάρχουν δύο τύποι διαθέσιμων agents: ο Zabbix agent (ελαφρύς, που υποστηρίζεται σε πολλές πλατφόρμες, γραμμένος σε C) και ο Zabbix agent 2 (εξαιρετικά ευέλικτος, εύκολα επεκτάσιμος με πρόσθετα, γραμμένος σε Go).[22][22]

5.4 Zabbix Server

Ο Zabbix server είναι η κεντρική διαδικασία του λογισμικού Zabbix.
Ο server εκτελεί τη δημοσκόπηση και την παγίδευση δεδομένων, στέλνει ειδοποιήσεις στους χρήστες. Είναι το κεντρικό στοιχείο στο οποίο οι zabbix agents αναφέρουν δεδομένα σχετικά με τη διαθεσιμότητα και την ακεραιότητα των συστημάτων. Ο server μπορεί ο ίδιος να ελέγχει εξ αποστάσεως υπηρεσίες δικτύου (όπως διακομιστές web και διακομιστές αλληλογραφίας) χρησιμοποιώντας απλούς ελέγχους υπηρεσιών[23].
Ο server είναι το κεντρικό αποθετήριο στο οποίο αποθηκεύονται όλες οι ρυθμίσεις παραμέτρων, τα στατιστικά και τα λειτουργικά δεδομένα και είναι η οντότητα στο Zabbix που θα ειδοποιεί ενεργά τους διαχειριστές όταν προκύπτουν προβλήματα σε οποιοδήποτε από τα συστήματα παρακολούθησης[23].
Η λειτουργία ενός βασικού server Zabbix χωρίζεται σε τρία ξεχωριστά στοιχεία.: server, web frontend και database storage.
Όλες οι πληροφορίες διαμόρφωσης για το Zabbix αποθηκεύονται στη βάση δεδομένων, με την οποία αλληλοεπιδρούν τόσο ο server όσο και το web frontend. Για παράδειγμα, όταν δημιουργείται ένα νέο στοιχείο χρησιμοποιώντας τη διεπαφή ιστού (ή API), προστίθεται στον πίνακα στοιχείων στη βάση δεδομένων. Στη συνέχεια, περίπου μία φορά το λεπτό ο διακομιστής Zabbix θα ζητήσει από τον πίνακα στοιχείων μια λίστα με τα στοιχεία που είναι ενεργά, η οποία στη συνέχεια αποθηκεύεται σε μια κρυφή μνήμη στον διακομιστή Zabbix. Αυτός είναι ο λόγος για τον οποίο μπορεί να χρειαστούν έως και δύο λεπτά για να εμφανιστούν οποιεσδήποτε αλλαγές έγιναν στο περιβάλλον του Zabbix[23].

5.5 Zabbix Agent

Ο Zabbix agent εφαρμόζεται στο σύστημα για την ενεργή παρακολούθηση των τοπικών πόρων και εφαρμογών (σκληροί δίσκοι, μνήμη, στατιστικά στοιχεία επεξεργαστή κ.λπ.).
Ο zabbix agent συλλέγει πληροφορίες τοπικά και αναφέρει δεδομένα στον διακομιστή Zabbix για περαιτέρω επεξεργασία. Σε περίπτωση αστοχιών (όπως η πλήρης φόρτωση του σκληρού δίσκου ή η κατάρρευση μιας διαδικασίας), ο διακομιστής Zabbix μπορεί να ειδοποιεί ενεργά τους διαχειριστές του συγκεκριμένου μηχανήματος που ανέφερε την αποτυχία[24].
Οι Zabbix agents μπορούν να εκτελούν παθητικούς και ενεργητικούς ελέγχους.
Σε έναν παθητικό έλεγχο ο agent απαντά σε ένα αίτημα δεδομένων. Ο Zabbix server (ή ο διακομιστής μεσολάβησης) ζητά δεδομένα, για παράδειγμα, φόρτωση CPU και ο Zabbix agent στέλνει πίσω το αποτέλεσμα.[24]
Οι ενεργοί έλεγχοι απαιτούν πιο περίπλοκη επεξεργασία. Ο agent πρέπει πρώτα να ανακτήσει μια λίστα στοιχείων από τον διακομιστή Zabbix για ανεξάρτητη επεξεργασία. Στη συνέχεια, θα στέλνει περιοδικά νέες τιμές στον διακομιστή[24].

5.6 Επίλογος

Από όσα έχουν διατυπωθεί, οδηγούμαστε στο συμπέρασμα ότι το Zabbix προσφέρει πολλές λειτουργίες που το κάνουν να είναι το κατάλληλο εργαλείο για την παρακολούθηση ενός server. Με

την εγκατάσταση του Zabbix server σε ένα τερματικό και την εγκατάσταση των Zabbix agents στα τερματικά που φιλοξενούν τα DBMS μπορούμε, εύκολα μέσα από το γραφικό περιβάλλον του Zabbix να ολοκληρώσουμε τον σκοπό παρακολούθησης όλων των MySQL/MariaDB servers.

Κεφάλαιο 6ο: Συμπεράσματα και προτάσεις βελτίωσης

Το θέμα αυτής της πτυχιακής είχε ως στόχο να βρει λύση στο πρόβλημα ημιαυτόματης παραγωγής αντιγράφου ασφαλείας και την παρακολούθηση λειτουργίας πολλαπλών servers MySQL και MariaDB.

Για το πρόβλημα του αντιγράφου ασφαλείας υπάρχουν έτοιμα εργαλεία που κάνουν αυτή τη δουλειά αλλά τα περισσότερα είναι με πληρωμή. Για αυτό το λόγο δημιουργήσαμε ένα shell script το οποίο δέχεται σαν παράμετρο μία λίστα με τους servers, συνδέεται με ssh σε κάθε έναν server και μέσω του εργαλείου mydumper παίρνει το αντίγραφο ασφαλείας. Για να μην διακόπτεται η διαδικασία, εφαρμόσαμε σε κάθε database server έναν χρήστη ο οποίος έχει την δυνατότητα σύνδεσης χωρίς κωδικό χρησιμοποιώντας το pluggable authentication. Με το crontab, που υπάρχει στο λειτουργικό σύστημα Debian Linux, καθορίζουμε τη χρονική περίοδο που θα εκτελείται το script. Το τελικό αποτέλεσμα είναι να έχουμε, στο τερματικό που εκτελεί την διαδικασία, τους φακέλους με τα αρχεία αντιγράφου ασφαλείας. Ο κάθε φάκελος περιέχει το όνομα του server και την ώρα που δημιουργήθηκε.

Ένας σημαντικός περιορισμός της λύσης αυτής είναι η διαδικασία προετοιμασίας. Θα ήταν προτιμότερο να δημιουργηθεί ένα ολοκληρωμένο λογισμικό, με γραφικό περιβάλλον, το οποίο με μία απλή εγκατάσταση ο χρήστης θα μπορεί να εισάγει τις διευθύνσεις των server που θέλει να πάρει αντίγραφο ασφαλείας. Ένα επιπλέον χαρακτηριστικό θα ήταν η αποστολή mail είτε όταν ολοκληρώνεται η διαδικασία είτε όταν κάτι πάει στραβά.

Όσον αφορά το κομμάτι της παρακολούθησης λειτουργίας κάθε server έγινε επιλογή ενός έτοιμου εργαλείου, του Zabbix. Το εργαλείο αυτό έχει την δυνατότητα να ενημερώνει τον διαχειριστή server, μέσα από γραφικό περιβάλλον, για τις μεταβλητές που αφορούν την λειτουργία του server, όπως ο χρόνος που είναι ενεργός, πόση κίνηση υπάρχει από χρήστες, κάποια βλάβη στο δίσκο ,κ.τ.λ.. Για την εγκατάσταση του Zabbix θα πρέπει να εγκατασταθεί ένας Zabbix server σε κάποιο τερματικό το οποίο καθορίζουμε ότι θα είναι το τερματικό παρακολούθησης και να εγκατασταθεί ένας Zabbix agent σε κάθε τερματικό που φιλοξενεί τον database server. Αυτό γίνεται γιατί οι agents μαζεύουν τη πληροφορία από την βάση δεδομένων, εκτελώντας ερωτήματα που επιστρέφουν μεταβλητές που αφορούν την κατάσταση της, και τα στέλνουν στον Zabbix server.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] "What Is a Database | Oracle." https://www.oracle.com/database/what-is-database/ (accessed Jul. 18, 2022).

[2] Paul Dubois, *Mysql (Developers Library)* , Fifth Edition.

[3] ben Fort, *MariaDB Crash Course* .

[4] Russell J.T. Dyer, *Learning MySQL and MariaDB*, First Edition. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

[5] "MySQL :: MySQL 8.0 Reference Manual :: 4.2.4 Connecting to the MySQL Server Using Command Options." https://dev.mysql.com/doc/refman/8.0/en/connecting.html (accessed Jul. 18, 2022).

[6] "MySQL :: MySQL 8.0 Reference Manual :: 6.2.17 Pluggable Authentication." https://dev.mysql.com/doc/refman/8.0/en/pluggable-authentication.html#pluggable-authentication-default-plugin (accessed Jul. 18, 2022).

[7] "MySQL :: MySQL 8.0 Reference Manual :: 6.4.1.10 Socket Peer-Credential Pluggable Authentication." https://dev.mysql.com/doc/refman/8.0/en/socket-pluggable-authentication.html (accessed Jul. 18, 2022).

[8] "MySQL :: MySQL 8.0 Reference Manual :: 7.1 Backup and Recovery Types." https://dev.mysql.com/doc/refman/8.0/en/backup-types.html (accessed Jul. 18, 2022).

[9] "MySQL :: MySQL 8.0 Reference Manual :: 4.5.4 mysqldump — A Database Backup Program." https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html (accessed Jul. 18, 2022).

[10] "GitHub - mydumper/mydumper: Official MyDumper project." https://github.com/mydumper/mydumper (accessed Jul. 18, 2022).

[11] "What Monitoring Is; Definition and Purpose." http://cec.vcn.bc.ca/cmp/modules/mon-wht.htm (accessed Jul. 18, 2022).

[12] "MySQL :: MySQL 5.7 Reference Manual :: 5.1.9 Server Status Variables." https://dev.mysql.com/doc/refman/5.7/en/server-status-variables.html (accessed Jul. 18, 2022).

[13] "MySQL :: MySQL 5.7 Reference Manual :: 25 MySQL Performance Schema." https://dev.mysql.com/doc/refman/5.7/en/performance-schema.html (accessed Jul. 18, 2022).

[14] "MySQL :: MySQL 5.7 Reference Manual :: 25.1 Performance Schema Quick Start." https://dev.mysql.com/doc/refman/5.7/en/performance-schema-quick-start.html (accessed Jul. 18, 2022).

[15] "MySQL :: MySQL 5.7 Reference Manual :: 26 MySQL sys Schema." https://dev.mysql.com/doc/refman/5.7/en/sys-schema.html (accessed Jul. 18, 2022).

[16] "MySQL :: MySQL Workbench." https://www.mysql.com/products/workbench/ (accessed Jul. 18, 2022).

[17] "Linux/UNIX: Bash Read a File Line By Line - nixCraft." https://www.cyberciti.biz/faq/unix-howto-read-line-by-line-from-file/ (accessed Jul. 19, 2022).

[18] “SSH Secure Shell home page, maintained by SSH protocol inventor Tatu Ylonen. SSH clients, servers, tutorials, how-tos.” https://www.ssh.com/academy/ssh#the-ssh-protocol (accessed Jul. 18, 2022).

[19] “Linux/Mac Tutorial: SSH Key-Based Authentication - How to SSH Without a Password - YouTube.” https://www.youtube.com/watch?v=vpk_1gldOAE&ab_channel=CoreySchafer (accessed Jul. 18, 2022).

[20] “How I use cron in Linux | Opensource.com.” https://opensource.com/article/17/11/how-use-cron-linux (accessed Jul. 18, 2022).

[21] “What is Zabbix.” https://www.zabbix.com/documentation/current/en/manual/introduction/about (accessed Jul. 18, 2022).

[22] “Zabbix overview.” https://www.zabbix.com/documentation/6.0/en/manual/introduction/overview (accessed Aug. 17, 2022).

[23] “Server.”https://www.zabbix.com/documentation/current/en/manual/concepts/server (accessed Jul. 18, 2022).

[24] “Agent.” https://www.zabbix.com/documentation/current/en/manual/concepts/agent (accessed Jul. 18, 2022).