ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Παρακολούθηση υποδομών με τη χρήση του εργαλείου Zabbix

Γεώργιος Π. Τσίρκας Α.Μ. 15318

Επιβλέπων καθηγητής: Τζήμας Ιωάννης

Αντίρριο – Μάιος 2019

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Αφιέρωση

Στην οικογένεια μου, που στηρίζει την προσπάθειά μου
να κάνω αυτό που θέλω και αγαπάω, όλα αυτά τα χρόνια…

Ευχαριστίες

Αρχικά θέλω να ευχαριστήσω τον συμφοιτητή μου Θωμά Καπούλα, ο οποίος με βοήθησε να μάθω αρκετά πράγματα για το αντικείμενο στο οποίο και εργάζομαι. Τους συναδέλφους μου από την εταιρεία modulus AE που έχουν αφιερώσει αρκετό χρόνο σε μένα, για να κατανοήσω γενικά την ιδέα του monitoring αλλά και την αρχιτεκτονική του εργαλείου Zabbix και να μπορέσω να εκπληρώσω την πτυχιακή μου εργασία.

Επίσης θέλω να ευχαριστήσω τον καθηγητή μου κ. Ιωάννη Τζήμα για την επίβλεψη της παρούσας πτυχιακής εργασίας.

Τέλος, θέλω να ευχαριστήσω την οικογένειά μου για την στήριξή τους όλα αυτά τα χρόνια.

Πρόλογος

Η αφοσίωση στην παρακολούθηση συστημάτων είναι το χαρακτηριστικό που κάνει έναν επαγγελματία διαχειριστή συστημάτων να ξεχωρίζει. Άπειροι διαχειριστές, δεν παρακολουθούν τα συστήματά τους, με αποτέλεσμα οι αποτυχίες να εντοπίζονται όταν κάποιος δυσαρεστημένος χρήστης καλέσει το helpdesk, για να εντοπίσει ή να λύσει το πρόβλημα που του έχει προκύψει. Πιο έντονο δε, παρατηρείται το φαινόμενο κατά το οποίο διαχειριστικές ομάδες εγκαθιστούν πλατφόρμες παρακολούθησης αλλά απενεργοποιούν τις "μετά από ώρες" ειδοποιήσεις διότι είναι αρκετά ενοχλητικές.

Αυτές οι τακτικές επηρεάζουν δυσμενώς μια εταιρεία, περιπλέκουν τις προσπάθειες επαναφοράς και δίνουν στην ομάδα των διαχειριστών συστημάτων μια κακή φήμη. Κάθε σύστημα προστίθεται στην πλατφόρμα παρακολούθησης πριν βγει στην παραγωγή υπηρεσιών και η μπάρα των ελέγχων στο γραφικό περιβάλλον ελέγχεται τακτικά και συντονισμένα. Οι μετρήσεις και οι τάσεις αξιολογούνται προληπτικά, έτσι τα προβλήματα μπορούν να εντοπιστούν πριν επηρεάσουν τους χρήστες ή θέσουν δεδομένα σε κίνδυνο. Ας μιλήσουμε για μια σημαντική υπηρεσία on-linevideostreaming. Ίσως έχετε ακούσει για τηλεμετρία. Η δικτυακή διαδικασία είναι τόσο σημαντική, που θα προτιμούσαν να διακοπεί η υπηρεσία παρά να διακοπεί το monitoring που κάνουν.

Χωρίς το monitoring , δεν θα είχαν ιδέα τι συμβαίνει στα συστήματα τους ή στις υπηρεσίες τους. Η φιλοσοφία του monitoring, κάνει έναν διαχειριστή συστημάτων υπερήρωα. Έστω ότι ανακαλύπτετε έναν καλύτερο τρόπο κατανόησης του λογισμικού σας, επιδιορθώνετε ορισμένα μικρά προβλήματα πριν πάρουν διαστάσεις και μεγαλώσουν, γίνεστε πιο αποτελεσματικοί, βρίσκοντας συνθήκες αποτυχίας και προβλήματα με τη τεχνική του debugging, και κατανοείτε τις επιδόσεις πολύπλοκων συστημάτων. Το monitoring βελτιώνει επίσης τον τρόπο ζωής, δίνοντάς σας τη δυνατότητα να επιδιορθώσετε προβλήματα σε σύντομο χρονικό διάστημα.

UseCase

Ας ξεκινήσουμε με μια αληθινή ιστορία για μια γνωστή εταιρεία. Η εταιρεία αυτή είχε έναν διαχειριστή. Αυτός διαχειρίζονταν όλη την υποδομή της εταιρείας που βρίσκονταν στα κέντρα

δεδομένων (datacenters). Κάθε φορά που η εταιρεία πρόσθετε ένα καινούριο διακομιστή, αυτός εγκαθιστούσε έναν agent παρακολούθησης και δημιουργούσε ορισμένους βασικούς ελέγχους για παρακολούθηση του συστήματος. Κάποιες φορές κάποιος από τους διακομιστές είχε πρόβλημα, το οποίο ενεργοποιούσε ειδοποιήσεις στέλνοντάς τες με sms και emails στον διαχειριστή. Αυτός με τη σειρά του έκανε τις ανάλογες διαδικασίες για να το επιδιορθώσει .
Για πολλά χρόνια αυτή η ενέργεια δούλευε αρκετά καλά. Βέβαια, υπήρχαν προβλήματα. Κάποιες φορές κάτι πήγαινε λάθος για το οποίο ενδεχομένως είτε να μην υπήρχε κάποιος έλεγχος, είτε δεν υπήρχε χρόνος να δράσει εγκαίρως ο διαχειριστής ή κάποιες υπηρεσίες που έτρεχαν στο μηχάνημα δεν παρακολουθούνταν καν. Αλλά σε μεγάλο βαθμό ή παρακολούθηση του συστήματος, ήταν καλή.

Αργότερα η IT βιομηχανία άρχισε να αλλάζει. Το virtualization και το cloud ήρθαν στο προσκήνιο και ο αριθμός των διακομιστών που χρειαζόταν να παρακολουθούνται αυξήθηκε. Κάποιοι από τους διακομιστές ήταν υπό τη διαχείριση ανθρώπων οι οποίοι δεν ήταν διαχειριστές συστημάτων, ή είχαν ανατεθεί σε τρίτους. Ορισμένοι από τους διακομιστές στο κέντρο δεδομένων (datacenter) μετακινήθηκαν στο cloud ή αντικαταστήθηκαν με software-as-a-service applications. Το πιο σημαντικό, είναι ότι η εταιρεία αυτή έγινε ένα κεντρικό κανάλι για επιχειρήσεις έτσι ώστε να επικοινωνούν και να πωλούν στους πελάτες τους. Προγράμματα και υπηρεσίες που πριν φαίνονταν να είναι απλώς τεχνολογία, τώρα έγιναν απαραίτητα για την ικανοποίηση των πελατών παρέχοντας υψηλής ποιότητας εξυπηρέτηση πελατών.
Ως αποτέλεσμα, πτυχές της παρακολούθησης συστημάτων ξεκίνησαν να χαλάνε. Ήταν δύσκολο να παρακολουθούνται οι διακομιστές, οι εφαρμογές και η υποδομή γινόντουσαν όλο και πιο περίπλοκες, οι προσδοκίες σχετικά με την διαθεσιμότητα και η ποιότητα έγιναν πιο απαιτητικές. Έγινε όλο και πιο δύσκολο να ελέγχονται όλες οι πιθανές καταστάσεις που θα μπορούσαν να πάνε στραβά χρησιμοποιώντας το συγκεκριμένο σύστημα.Οι ειδοποιήσεις είχαν συσσωρευτεί. Περισσότεροι διακομιστές και υπηρεσίες σήμαιναν περισσότερη ζήτηση για συστήματα παρακολούθησης, εκ των οποίων τα πιο πολλά ήταν σε θέση να αντικατασταθούν με την προσθήκη μεγαλύτερων και πιο δυνατών διακομιστώνκαι δεν μπορούσαν εύκολα να κατανεμηθούν. Κάτω από τέτοιο υψηλό φόρτο επεξεργαστικής ισχύος και μνήμης , η ανίχνευση και ο εντοπισμός βλαβών γινόντουσαν όλο και πιο αργοί και δύσκολοι.

Η επιχείρηση ξεκίνησε να απαιτεί περισσότερα δεδομένα, τόσο για να αναδείξει την ποιότητα των υπηρεσιών που προσφέρει στους πελάτες της, όσο και για να δικαιολογήσει την αύξηση των δαπανών για IT υπηρεσίες. Πολλές από αυτές τις απαιτήσεις έγιναν για δεδομένα τα οποία η υπάρχουσα παρακολούθηση απλώς δεν μετρούσε ή δεν μπορούσε να δημιουργήσει. Το σύστημα παρακολούθησης είχε γίνει υπερβολικά περίπλοκο. Για πολλούς διαχειριστές συστημάτων αυτή η κατάσταση monitoring δεν ήταν αποδοτική. Δεν υπάρχει λόγος να παραμείνει έτσι, καθώς υπάρχει καλύτερη λύση, η οποία απευθύνεται στην αλλαγή του τρόπου λειτουργίας του IT και η οποία θα κάνει το IT τμήμα να είναι πιο λειτουργικό και μελλοντικά επεκτάσιμο.

Περίληψη

Το θέμα της διπλωματικής εργασίας σχετίζεται με την παρακολούθηση συστημάτων και το εργαλείο παρακολούθησης Zabbix. Η παρακολούθηση συστημάτων και προγραμμάτων είναι η πιο σημαντική λειτουργία που πρέπει να αναπτύσσεται και να εξελίσσεται σε οποιαδήποτε υποδομή, καθώς μπορούν να αποτραπούν ζημιές τεράστιου κόστους που κατά καιρούς έχουν συμβεί σε μεγάλες και ευρέως γνωστές εταιρείες. Το περιεχόμενο της εργασίας παρουσιάζει γενικά την παρακολούθηση συστημάτων και προγραμμάτων τόσο στον IT τομέα όσο και στον εμπορικό, αναλύοντας μοντέλα και λειτουργίες. Αναφέρεται στα ευρέως γνωστά εργαλεία παρακολούθησης. Λόγω του ότι τα τελευταία τέσσερα χρόνια οι βάσεις δεδομένων χρονοσειρών έχει βελτιωθεί αρκετά ως προς την ταχύτητα αναζήτησης δεδομένων και τη χρήση λιγότερων υπολογιστικών πόρων, σε σχέση με τις άλλες βάσεις δεδομένων, είναι ιδανικές στην παρακολούθηση συστημάτων. Γι αυτό, έχουν αρχίσει να γίνονται ενσωματώσεις με διάφορα εργαλεία παρακολούθησης όπως το zabbix. Ωστόσο, στη συνέχεια, απευθύνεται στις βάσεις δεδομένων χρονοσειρών. Ακολούθως, αναπτύσσεται το εργαλείο zabbix (αρχιτεκτονική, ροή δεδομένων, χαρακτηριστικά κλπ.). Τέλος, συμπεραίνεται ο λόγος για τον οποίο είναι ένα ιδανικό εργαλείο για παρακολούθηση συστημάτων και προγραμμάτων.

Monitoring

Τι είναι το Monitoring;

Από τεχνολογική άποψη, το monitoring είναι τα εργαλεία και οι λειτουργίες με τις οποίες μετράμε και διαχειριζόμαστε τα IT συστήματα. Το monitoring όμως είναι κάτι παραπάνω από αυτό. Μεταφράζει τις μετρικές που παρήχθησαν από τα συστήματα ή από προγράμματα σε πολύτιμα δεδομένα για μια επιχείρηση. Αυτά τα δεδομένα παρέχουν ένα feedback στην επιχείρηση που βοηθά να διασφαλιστεί ότι παρέχεται αυτό που θέλει ο πελάτης. Τα δεδομένα αυτά επίσης παρέχουν ένα feedback στο IT τμήμα για να υποδείξει τι δεν δουλεύει και τι είναι αυτό που παρέχει ανεπαρκή ποιότητα υπηρεσιών.

Το monitoring στον επιχειρησιακό τομέα

Tomonitoring υπάρχει για να βοηθάει μια επιχείρηση και να εξασφαλίζει τη σωστή λειτουργία της. Παρέχει δεδομένα, τα οποία της επιτρέπουν να κάνει σωστές επενδύσεις σε προϊόντα και σε νέες τεχνολογίες που θα αποφέρουν μεγαλύτερα κέρδη.

Το monitoring στον IT τομέα

Πλέον όλα τα IT τμήματα των επιχειρήσεων βασίζονται στο monitoring το οποίο τους επιτρέπει να έχουν γνώση για την κατάσταση των applications, των hosts και γενικά της ευρύτερης τεχνολογικής υποδομής της επιχείρησης. Με αυτό μπορούν να εντοπίζουν, να κάνουν διάγνωση και να επιλύουν σφάλματα και άλλα προβλήματα. Το monitoring είναι αυτό που παρέχει ένα μεγάλο μέρος δεδομένων τα οποία προσφέρουν γνώση για τις κρίσιμες αποφάσεις πάνω σε προϊόντα και τεχνολογία, και μετράει την επιτυχία αυτών των projects. Είναι ένα σημαντικό

κομμάτι για τον κύκλο ζωής της διαχείρισης ενός προϊόντος, για την σχέση με τους εσωτερικούς πελάτες και βοηθά να αποδειχθεί ότι τα χρήματα μιας επιχείρησης επενδύονται σωστά.

Μοντέλα του Monitoring

Κάθε εταιρεία χρησιμοποιεί ένα από τα δύο ή και τα δύο μοντέλα monitoring ανάλογα με τις ανάγκες της, τη γνώση και την εμπειρία του προσωπικού. Τα μοντέλα αυτά είναι τα εξής:

1. Προληπτικό μοντέλο
2. Αντιδραστικό μοντέλο

Προληπτικό μοντέλο

Με το μοντέλο αυτό οι ομάδες IT ελέγχουν την συμπεριφορά και την κατάσταση των υπηρεσιών καθώς τρέχουν και τον εντοπισμό πιθανών κινδύνων, 7 ημέρες την εβδομάδα και 24 ώρες τη μέρα. Τα μέλη της ομάδας IT χρησιμοποιούν εργαλεία παρακολούθησης (zabbix, nagios, observium, prometheus) για να βλέπουν την τρέχουσα κατάσταση, να προγραμματίζουν δράσεις για την αποφυγή πιθανών προβλημάτων και να συγκρίνουν τις επιδόσεις υπηρεσιών ή συστημάτων με άλλα αντίστοιχα. Κάποια από τα πλεονεκτήματα του προληπτικού monitoring είναι:

- Γρήγορος εντοπισμός προβλημάτων
- Συνεχής παρακολούθηση συστημάτων

Αντιδραστικό μοντέλο

Το αντιδραστικό monitoring είναι κυρίως αυτόματο με κάποια κατάλοιπα στοιχεία τα οποία δεν παρακολουθούνται. Έχουν αξιοποιηθεί εργαλεία υψηλής πολυπλοκότητας για την πραγματοποίηση παρακολούθησης συστημάτων. Με κάποια κοινά εργαλεία όπως Zabbix ή Prometheus γίνεται έλεγχος βασικών λειτουργιών όπως δίσκος, επεξεργαστής και μνήμη. Συλλέγονται κάποια δεδομένα και οι ειδοποιήσεις βασίζονται σε κοινές σταθερές, στέλνονται με email ή αλλες υπηρεσίες μηνυμάτων και υπάρχουν μία ή περισσότερες κονσόλες απεικονίζοντας την κατάσταση παρακολούθησης. Tomonitoring παρέχει κάποια δεδομένα τα οποία μετρούν την

ποιότητα ή τις υπηρεσίες και άλλα δεδομένα τα οποία βοηθούν την επιχείρηση να δικαιολογεί προϋπολογισμούς, κόστη και νέα projects. Τα περισσότερα από αυτά τα δεδομένα πρέπει να μετασχηματιστούν πριν χρησιμοποιηθούν. Τυπικά, το αντιδραστικό monitoring κατασκευάζεται και αναπτύσσεται από τους διοικούντες μιας επιχείρησης. Οι ενημερώσεις στα συστήματα παρακολούθησης τείνουν να αντιδράσουν σε περιστατικά και διακοπές υπηρεσιών. Οι νέοι έλεγχοι είναι, συνήθως, το τελευταίο βήμα στην ανάπτυξη εφαρμογών ή υποδομών.

Blackbox και Whiteboxmonitoring

Τα περισσότερα εργαλεία παρακολούθησης χρησιμοποιούν δύο τεχνικές monitoring, την blackbox και whitebox. Η τεχνική blackbox ελέγχει το πρόγραμμα ή την υπηρεσία από έξω, δηλαδή ρωτάς το σύστημα για κάποια χαρακτηριστικά της υπηρεσίας, π.χ. αν ακούει σε κάποια συγκεκριμένη πόρτα και να επιστρέφει τα σωστά δεδομένα ή returncode. Πολλά συστήματα χρησιμοποιούν την τεχνική αυτή επειδή δεν έχουν φτιαχτεί λειτουργίες όπου να είναι εύκολη η παρακολούθηση.

Η τεχνική του whitebox από την άλλη ελέγχει εσωτερικά το πρόγραμμα ή την υπηρεσία. Κατά την εκκίνηση ενός προγράμματος, το ίδιο το πρόγραμμα επιστρέφει την κατάσταση στην οποία βρίσκεται, τις επιδόσεις για τις λειτουργίες που κάνει και τους υπολογιστικούς πόρους που χρησιμοποιεί. Η whitebox παρακολούθηση είναι εφικτή είτε αποevents και logs που διανέμονται στο εργαλείο παρακολούθησης είτε το πρόγραμμα εκθέτει πληροφορίες, τις οποίες μπορεί να αποσπάσει ένα εργαλείο παρακολούθησης.

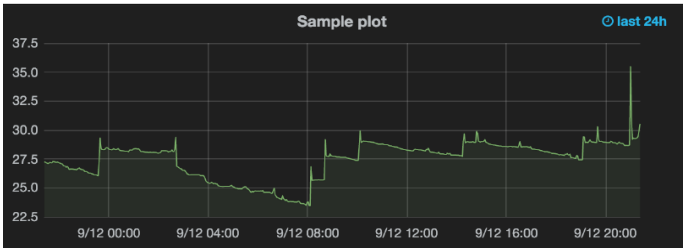
Τύποι δεδομένων

Τα δεδομένα που συλλέγονται για την παρακολούθηση συστημάτων χωρίζονται σε τρεις κατηγορίες:

- Μετρικές: Χαρακτηρίζουν την λειτουργική κατάσταση ενός συστήματος, υπηρεσίας, κλπ. Τυπικά είναι τιμές αριθμών ή Boolean.
- Γεγονότα: Δείχνουν τις αλλαγές που μπορεί να συμβούν στο σύστημα.
- Logs: Είναι ένα υποσύνολο γεγονότων. Μας βοηθάνε να δούμε τι συμβαίνει στο σύστημα ή σε μια εφαρμογή και είναι αρκετά χρήσιμα σε διάγνωση και έρευνα σφαλμάτων.

Μετρικές και τύποι

Οι μετρικές είναι μετρήσεις των υπαρχόντων υλικού και λογισμικού. Για να κάνετε μια μετρική χρήσιμη, παρακολουθείτε την κατάσταση της και καταγράφετε σημεία δεδομένων. Ο συνδυασμός αυτών των τύπων δεδομένων ονομάζεται σειρά χρόνου. Ένα κλασικό παράδειγμα μιας μετρικής που ίσως παρακολουθείτε σε σειρά χρόνου είναι οι προβολές μιας ιστοσελίδας. Συνηθίζεται να γίνεται συλλογή δεδομένων σε συγκεκριμένο μεσοδιάστημα. Το μεσοδιάστημα αυτό μπορεί να είναι από ένα δευτερόλεπτο έως πέντε λεπτά ή και μια ώρα.



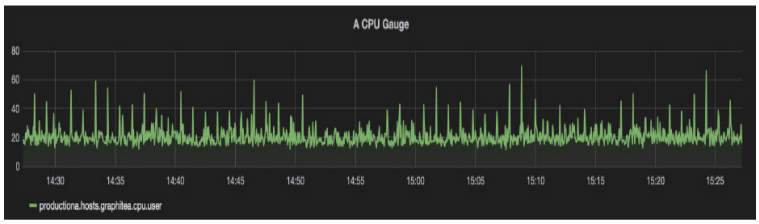
Εικονα 1.1

Τα γραφήματα μπορούν να είναι αρκετά χρήσιμα για εμάς. Παρέχουν μια απεικόνιση των κρίσιμων δεδομένων, τα οποία είναι σχετικά εύκολο να γίνουν αντιληπτά, τουλάχιστον με περισσότερη ευκολία από το να διαβαστούν τα ίδια δεδομένα σε μια φόρμα από λίστες τιμών. Επίσης, το γράφημα απεικονίζει τα δεδομένα δείχνοντας το ιστορικό, το οποίο μας αναφέρει τι έχει αλλάξει και πότε. Μπορούμε να χρησιμοποιήσουμε αυτά τα χαρακτηριστικά για να καταλάβουμε τι συμβαίνει ή τι είχε συμβεί στο σύστημά μας. Ένα απλό γράφημα απεικονίζεται στην εικόνα 1.1

Υπάρχουν διάφοροι τύποι μετρικών που θα δείτε σε κάποιο monitoring εργαλείο.

Gauges

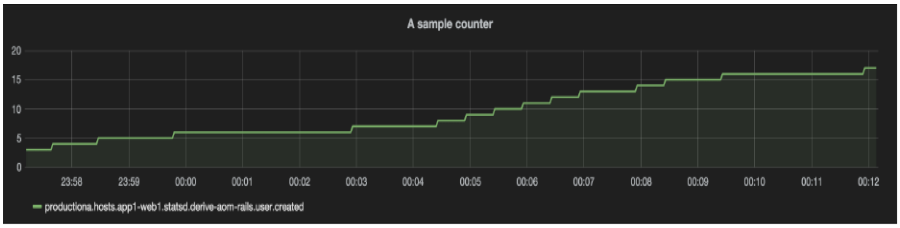
Ο πρώτος και πιο συνηθισμένος τύπος μετρικής που θα δούμε είναι ένα gauge. Τα gauges είναι αριθμοί που αναμένονται να αλλάζουν με την πάροδο του χρόνου. Ένα γράφημα gauge απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 1.2

Counters

Ο δεύτερος τύπος μετρικής που μπορεί να συναντάτε συχνά είναι ένας counter. Ο τύπος αυτός είναι αριθμοί οι οποίοι αυξάνονται με την πάροδο του χρόνου και ποτέ δεν μειώνονται. Κάποιες φορές μπορεί να μηδενίσουν και κατά την επανεκκίνηση αυξάνονται. Παραδείγματα καταστάσεων που χρησιμοποιούμε counter σε προγράμματα είναι το systemuptime, bytes που έχουμε στείλει και λάβει σε μια συσκευή. Αντίστοιχο παράδειγμα για επιχειρήσεις είναι ο αριθμός πωλήσεων σε ένα μήνα ή το κόστος πωλήσεων για μια χρονική περίοδο. Ένα γράφημα counter απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 1.3

Timers

Οι timers παρακολουθούν πόσο διαρκεί ένα γεγονός. Χρησιμοποιούνται συνήθως για παρακολούθηση προγραμμάτων, π.χ. ενσωμάτωση ενός timer στη διαδικασία ενός backup. Έτσι, μετράνε πόσο χρόνο πήρε για να εκτελεστεί όλη η διαδικασία του backup. Ένα γράφημα ενός counter απεικονίζεται στην παρακάτω εικόνα.

Εικόνα 1.4

Πολλές φορές η τιμή μιας μετρικής δεν μας είναι χρήσιμη. Για το λόγο αυτό, η απεικόνιση μιας μετρικής απαιτεί την εφαρμογή μαθηματικών πράξεων. Για παράδειγμα, μπορεί να θέλουμε στατιστικές λειτουργίες. Κάποιες από αυτές είναι:

- Count: Μετράει τον αριθμό των δεδομένων που έχουμε συλλέξει σε ένα συγκεκριμένο μεσοδιάστημα
- Sum: Προσθέτει τις τιμές των δεδομένων που έχουμε συλλέξει σε ένα συγκεκριμένο μεσοδιάστημα
- Average: Ο μέσος όρος όλων τιμών σε ένα συγκεκριμένο μεσοδιάστημα.
- Median: Η διάμεσος είναι το κέντρο όλων των τιμών, όπου το 50% θα είναι από πάνω και το άλλο 50% θα είναι κάτω από αυτό.
- Percentiles: Μετράει το ποσοστό των τιμών που λαμβάνουμε.
- Standard Deviation: Υπολογίζει το ποσό της μεταβολής ενός συνόλου τιμών δεδομένων. Η τυπική απόκλιση του 0 σημαίνει ότι το σύνολο των τιμών είναι ίσο με το μέσο όρο των δεδομένων. Μεγαλύτερες αποκλίσεις, σημαίνουν ότι τα δεδομένα απλώνονται πάνω από το εύρος των τιμών.
- RatesofChange: Αναπαραστούν την αλλαγή που έχει γίνει μεταξύ των δεδομένων σε μια χρονική σειρά.
- Frequencydistributions: Ξεκινάει με τη διαδικασία συγκέντρωσης δεδομένων, γνωστή και ως binning, και παρουσιάζει τα δεδομένα με τέτοιο τρόπο έτσι ώστε τα σχετικά μεγέθη τιμών να απεικονίζονται.

Visualization

Η απεικόνιση δεδομένων είναι μια απίστευτα πανίσχυρη τεχνική ανάλυσης και ερμηνείας καθώς και ένα εξαιρετικό εργαλείο για να μάθει κάποιος. Ωστόσο, η απεικόνιση των μετρικών είναι συχνά δύσκολο να ερμηνευτεί. Οπότε οι άνθρωποι να αντιλαμβάνονται τις σχέσεις και το νόημα δύο άσχετων πραγμάτων ή σημαντικά μοτίβα με τυχαία δεδομένα όταν βλέπουν ορισμένα γραφήματα. Αυτό οδηγεί σε ξαφνικά άλματα από τη συσχέτιση σε αιτιώδη συνάφεια και μπορεί να επιδεινωθεί από την λεπτομέρεια και την ανάλυση των διαθέσιμων δεδομένων στο πως να διαλέξουμε να το παρουσιάσουμε και σε ποιον θα το παρουσιάσουμε.
Οι ιδανικές απεικονίσεις πρέπει να είναι ξεκάθαρες και να δείχνουν τα δεδομένα δίνοντας έμφαση στις επισημάνσεις. Ωστόσο θα πρέπει να ακολουθούνται κάποιοι κανόνες όπως:

- Να φαίνονται ξεκάθαρα τα δεδομένα
- Να κάνει τον αναγνώστη - θεατή να διαβάσει την ουσία και όχι ότι φαίνεται.
- Να αποφεύγεται η παραμόρφωση των δεδομένων
- Να καθίσταται συναφής μεγάλος όγκος δεδομένων

Notifications

Οι ειδοποιήσεις είναι η πρωτεύουσα έξοδος στο monitoring. Μπορεί να αποτελείται από emails, SMS, αναδυόμενα παράθυρα, στιγμιαία μηνύματα στον υπολογιστή μας ή οποιαδήποτε άλλη μέθοδο, προκειμένου να γνωρίζουμε τις αλλαγές που γίνονται στα συστήματαά μας. Ίσως να φαίνεται πολύ απλό αλλά υπάρχει μεγάλη πολυπλοκότητα, κακή εφαρμογή και διαχείριση. Για να υλοποιήσετε ένα καλό σύστημα ειδοποιήσεων, χρειάζεται να σκεφτείτε τα εξής:

- Σε ποιους να σταλεί η ειδοποίηση σχετικά με το πρόβλημα.
- Πως να τους το πει
- Πόσο συχνά να τους το λέει
- Όταν σταματήσει να τους το λέει, να κάνει κάτι άλλο, ή να το μεταβιβάσει σε κάποιον άλλο.

Αν κάνετε λάθος και δημιουργήσετε πάρα πολλές ειδοποιήσεις, τότε οι χρήστες δεν θα μπορούν να αναλάβουν όλα τα προβλήματα και θα τις βάλουν σε σίγαση. Όλοι έχουμε κακές ιστορίες από φακέλους στο inbox που έχουν γεμίσει με αδιάβαστα μηνύματα ειδοποιήσεων από συστήματα monitoring. Κάποιες φορές δημιουργούνται τόσες πολλές ειδοποιήσεις πους τις αγνοούμε και κάποιες φορές τις διαγράφουμε. Αυτό έχει ως αποτέλεσμα να χάσουμε κρίσιμες ειδοποιήσεις. Το πιο σημαντικό είναι ότι χρειάζεται επεξεργασία του περιεχομένου που θα λαμβάνει ο παραλήπτης της ειδοποίησης. Οι ειδοποιήσεις είναι συνήθως το μοναδικό σημάδι που λαμβάνεις και σε ενημερώνει ότι κάτι υπολειτουργεί ή χρειάζεται προσοχή. Πρέπει να είναι συνοπτικές, σωστά διατυπωμένες, ακριβείς και ευκολομνημόνευτες. Σχεδιάζοντας τις ειδοποιήσεις στην πραγματικότητα είναι χρήσιμο και κρίσιμο.

Υπάρχει μία μεγάλη ποικιλία από προγράμματα τόσο σε επίπεδο selfhosted όσο και σε επίπεδο Cloud των οποίων η δουλειά είναι να συλλέγουν δεδομένα από διάφορες πηγές όπως λειτουργικά συστήματα, webapplications ή ακόμα και βάσεις δεδομένων τα οποία επεξεργάζονται, δημιουργούν events, triggers και απεικονίσεις. Κάποια από αυτά είναι το Nagios, το Observium, το Promitheus, το Datatog, το Zabbix και πολλά άλλα.
Στη συνέχεια θα μιλήσουμε για το Zabbix, ένα από τα κορυφαία monitoringapplications που έχει βελτιωθεί σημαντικά την τελευταία πενταετία.

Εργαλεία παρακολούθησης

Τα εργαλεία παρακολούθησης δεν παρέχουν μόνο λεπτομερείς πληροφορίες σχετικά με την υγεία των διακομιστών σας, αλλά περιλαμβάνουν επίσης και benchmarking επιδόσεων, δυνατότητες προειδοποίησης, λεπτομερή αναφορά, απεικόνιση δεδομένων και ούτω καθεξής, διατηρώντας τις εφαρμογές ή τις υπηρεσίες σας. Με τόσα πολλά εργαλεία, με αμέτρητα διαθέσιμα χαρακτηριστικά, είναι δύσκολο να αποφασίσετε για το σωστό εργαλείο για το αντικείμενο που θέλετε. Για παράδειγμα, στο στάδιο του κύκλου ζωής του προϊόντος, κάποιος ψάχνει για ένα εργαλείο χαμηλού κόστους και εύκολης ανάπτυξης με βασικές δυνατότητες παρακολούθησης διακομιστών και εφαρμογών. Καθώς το προϊόν κλιμακώνεται σε ετερογενείς αρχιτεκτονικές ανάπτυξης, αναζητάτε μια ολοκληρωμένη λύση προσαρμόσιμη με καλές

δυνατότητες αναφοράς και απεικόνισης. Τα εργαλεία που είναι πλούσια σε χαρακτηριστικά και προσαρμόσιμα τείνουν να είναι πιο πολύπλοκα και απαιτούν μια ειδική ομάδα DevOps για τη ρύθμιση και τη διαχείριση των εργαλείων. Αυτή η αναφορά συγκρίνει την ικανότητα έναντι ευκολίας υιοθέτησης μερικών από τα δημοφιλέστερα εργαλεία παρακολούθησης.

Δυνατότητες

Η βαθμολογία δυνατοτήτων δεν λαμβάνει υπόψη μόνο τα διαφορετικά χαρακτηριστικά που υποστηρίζονται από το εργαλείο αλλά και απαιτήσεις όπως προσαρμοστικότητα, κλιμάκωση και απόδοση.

Πλήρη χαρακτηριστικά

Τα ακόλουθα χαρακτηριστικά θεωρούνται μέτρα σύγκρισης.

- Παρακολούθηση των υποδομών
- Παρακολούθηση εφαρμογών
- Πίνακας ελέγχου
- Υποστήριξη HPC
- Υποστήριξη Αρχιτεκτονικής Μικρών Υπηρεσιών
- Ενημέρωση Δημόσιας παρακολούθησης CloudCustomizability

Προσαρμοστικότητα

Η προσαρμοστικότητα ενός εργαλείου μετράται από τη δυνατότητα να προσδιορίσει κάποιος μια νέα μέτρηση ή να τροποποιήσει μια υπάρχουσα.

Κλιμακωσιμότητα

Η κλιμακωσιμότητα αναφέρεται στο πόσο καλά το εργαλείο παρακολούθησης μπορεί να κλιμακωθεί υπό φορτίο (αυξημένος αριθμός εξυπηρετητών).

Ευκολία υιοθεσίας

Η ευκολία υιοθεσίας αποτελεί βασικό παράγοντα στην επιλογή ενός εργαλείου για τις νεοσύστατες επιχειρήσεις. Η ευκολία υιοθέτησης είναι μέτρο του χρόνου που απαιτείται για την

εκμάθηση, υιοθέτηση, εφαρμογή και συντήρηση του εργαλείου. Το εύρος ευκολίας υιοθέτησης λαμβάνει υπόψη τον αριθμό των συνεισφερόντων του Github, τις εκδόσεις σε ένα χρόνο, τον χρόνο που απαιτείται για την ανάπτυξη και τη διαμόρφωση του εργαλείου και το κόστος αδειοδότησης.

Το χρονικό διάστημα που λήφθηκε για την ανάπτυξη και τη διαμόρφωση

Το χρονικό διάστημα που απαιτείται για την ανάπτυξη και τη διαμόρφωση του εργαλείου παρακολούθησης αποτελεί σημαντικό παράγοντα, καθώς ο χρόνος κυμαίνεται από μία έως τρεις ώρες.

Κάποια εργαλεία παρακολούθησης είναι δωρεάν και κάποια επί πληρωμή. Άλλα είναι cloud και άλλα selfhosted. Κάποια είναι μόνο για δικτυακή υποδομή, ενώ άλλα είναι για servers, cloud κλπ.

Παρακάτω, αναφέρονται τα πιο δημοφιλή εργαλεία παρακολούθησης:

SolarWinds

Το SolarWinds είναι ένα λογισμικό παρακολούθησης απόδοσης IT / Networking. Παρακολουθεί το περιβάλλον δικτύου σας και εντοπίζει και επιλύει ζητήματα που παρεμποδίζουν την απόδοση του δικτύου. Το SolarWinds επικεντρώνεται στους επαγγελματίες τεχνολογιών πληροφορικής, MSP και DevOps και εξαλείφει την πολυπλοκότητα των παραδοσιακών προμηθευτών λογισμικού επιχειρήσεων. Είναι επί πληρωμή και εγκαθίσταται μόνο σε λειτουργικό Windows

Cacti

Είναι μια λύση γραφικής παράστασης δικτύου. Παρέχει ένα γρήγορο, προχωρημένο σχεδιασμό γραφημάτων, πολλαπλές μεθόδους απόκτησης δεδομένων και αποκλειστικές δυνατότητες διαχείρισης χρηστών. Είναι εύκολο να αναπτυχθεί και να υποστηριχθεί η βασική παρακολούθηση δικτύου και διακομιστών. Είναι προτιμότερο όταν ο αριθμός των διακομιστών είναι χαμηλός.

Nagios

Το Nagios είναι ένα δημοφιλές εργαλείο παρακολούθησης ανοιχτού κώδικα. Μπορεί να χρησιμοποιηθεί τόσο για παρακολούθηση διακομιστών όσο και εφαρμογών. Είναι εύκολο να διαμορφώσετε και να προσαρμόσετε σε κάποιο βαθμό. Σε περίπτωση αποτυχίας, το Nagios μπορεί να προειδοποιήσει το τεχνικό προσωπικό του προβλήματος, επιτρέποντάς του να ξεκινήσει διαδικασίες αποκατάστασης προτού οι διακοπές επηρεάσουν την επιχείρηση.

Zabbix

Το Zabbix προσφέρει εξαιρετική απόδοση για συλλογή δεδομένων και μπορεί να κλιμακωθεί σε πολύ μεγάλα περιβάλλοντα. Οι κατανεμημένες επιλογές παρακολούθησης διατίθενται επίσης με χρήση των proxyZabbix. Το Zabbix περιλαμβάνει ένα web-basedinterface, ασφαλή έλεγχο ταυτότητας χρήστη και ένα ευέλικτο σχήμα δικαιωμάτων χρήστη.

Icinga

Το Icinga είναι ένα δημοφιλές εργαλείο παρακολούθησης ανοιχτού κώδικα που ελέγχει τους κεντρικούς υπολογιστές και τις υπηρεσίες και σας ενημερώνει για τις καταστάσεις τους. Το Icinga είναι ένα fork του Nagios και είναι συμβατό και με πιο παλαιά συστήματα. Αυτό το εργαλείο έχει αποκτήσει δημοτικότητα λόγω της κοινοτικής υποστήριξης που έχει όσον αφορά τις συνεισφορές Github. Είναι η καλύτερη επιλογή χαμηλού κόστους για ένα προϊόν πρώιμης φάσης. Το Icinga διαθέτει καλά χαρακτηριστικά επιβίβασης στην ταμπέλα και είναι εύκολο να αναπτυχθεί. Δεν συστήνεται για μικροϋπηρεσίες ή υποδομή HPC.

Riemann

Το Riemann συγκεντρώνει τα γεγονότα από διακομιστές και εφαρμογές με ισχυρή ρέουσα γλώσσα επεξεργασίας. Είναι αεικίνητο και δεν παρέχει κάποιο καινοτόμο μέσο αποθήκευσης για τα γεγονότα. Μπορεί να ενσωματωθεί σε οποιαδήποτε βάση δεδομένων χρονοσειρών όπως pipelineDB / InfluxDB για την αποθήκευση ιστορικών δεδομένων και με Graphite και Grafanaγια πίνακες ελέγχου. Το API του είναι σε Clojure. Είναι ιδανικό για υποδομές HPC και μικροϋπηρεσίες. Είναι η πιο σύγχρονη τεχνολογία και απαιτεί εξειδικευμένη ομάδα DevOps να κάνει τις προσαρμογές.

Prometheus

Ο Prometheus είναι ένα σύστημα παρακολούθησης συστημάτων και υπηρεσιών. Συλλέγει μετρήσεις από τους διαμορφωμένους στόχους σε συγκεκριμένα χρονικά διαστήματα, αξιολογεί τις εκφράσεις κανόνων, εμφανίζει τα αποτελέσματα και μπορεί να ενεργοποιήσει ειδοποιήσεις εάν κάποια κατάσταση παρατηρηθεί ότι είναι αληθής. Η ευέλικτη γλώσσα της ερώτησης μας επιτρέπει να συγκεντρώνουμε δεδομένα σε όλες τις διαστάσεις. Είναι κατάλληλο για προϊόντα που αναπτύσσονται σε διαχειριστές cluster, όπως είναι τα εργαλεία παρακολούθησης.

DataDog

Η Datadog είναι μια πλατφόρμα παρακολούθησης και ανάλυσης βασισμένη σε SaaS για παρακολούθηση διακομιστών, βάσεων δεδομένων και εφαρμογών. Παρέχει μια ενιαία προβολή των εφαρμογών που εκτελούνται σε κλίμακα στο cloud. Το Datadog χρησιμοποιεί έναν πράκτορα ανοιχτού κώδικα γραμμένο στη Python για τη συγκέντρωση μετρήσεων και συμβάντων. Η καμπύλη χαμηλής εκμάθησης και οι δυνατότητες του κουτιού καθιστούν ευκολότερη και ταχύτερη τη λειτουργία του. Είναι κατάλληλη για τις εταιρείες πρώιμων προϊόντων, χωρίς μια ομάδα εξειδικευμένων συνεργείων. Το κόστος θα μπορούσε να είναι υψηλό σε μεγάλες υποδομές.

NewRelic

Το νέο Relic είναι λύση διαχείρισης απόδοσης που επιτρέπει στους προγραμματιστές να εντοπίζουν και να διορθώνουν προβλήματα απόδοσης εφαρμογών σε πραγματικό χρόνο. Τα χαρακτηριστικά επιβίβασης στην ταμπέλα είναι επίσης πολύ καλά.

Ποια είναι τα βασικά στοιχεία που πρέπει να παρακολουθούνται σε μια υποδομή;

Παρακολούθηση CPU - Τα κύρια στοιχεία CPU που είναι ιδανικό να παρακολουθείτε είναι:

- CPU load - αντιπροσωπεύει το μέσο φορτίο της CPU για μια χρονική περίοδο. Συμβολικά εμφανίζεται με τη μορφή τριών αριθμών που αντιπροσωπεύουν το φορτίο του συστήματος κατά τη διάρκεια των τελευταίων περιόδων ενός, πέντε και δεκαπέντε λεπτών.

- CPU systemtime - είναι ο χρόνος που ο CPU είναι απασχολημένος με την εκτέλεση κώδικα στον χώρο του πυρήνα.
- CPU usertime - είναι ο χρόνος που ο CPU είναι απασχολημένος στην εκτέλεση κώδικα στο χώρο του χρήστη.
- CPU iowaittime - είναι μια υποκατηγορία του CPU idletime. Επομένως, όσο υπάρχει άλλη διαδικασία που μπορεί να επεξεργαστεί η CPU, θα το κάνει αυτό. Είναι βασική μέτρηση για να γνωρίζετε έναν ανεβασμένο IO.
- CPU nicetime- ο χρόνος που η CPU έχει δαπανήσει για τη διεξαγωγή διαδικασιών χρηστών που έχουν γίνει "nice".
- CPU stealtime - σε εικονικό υλικό είναι ο χρόνος που το λειτουργικό σύστημα θέλησε να εκτελέσει κάτι αλλά δεν του επιτράπηκε από τον hypervisor. Αυτό μπορεί να συμβεί αν το φυσικό υλικό εκτελεί πολλαπλά λειτουργικά συστήματα επισκεπτών και ο hypervisor έχει επιλέξει να διαθέσει ένα μέρος CPU σε ένα άλλο.
- CPU idletime - είναι ο χρόνος που ο CPU δεν ήταν απασχολημένος. Ο χρόνος αδράνειας μετράει την αχρησιμοποίητη χωρητικότητα της CPU.
- CPU interrupttime

Παρακολούθηση Μνήμης - Τα κύρια στοιχεία μνήμης που είναι ιδανικό να παρακολουθείτε είναι:

- Διαθέσιμη μνήμη
- Συνολική Μνήμη
- Συνολικός χώρος swap
- Ελεύθερος χώρος swap
- Ποσοστό ελεύθερου χώρου swap

Παρακολούθηση δίσκου - Τα κύρια στοιχεία δίσκου που είναι ιδανικό να παρακολουθείτε είναι:

- Διαθέσιμος χώρος
- Ποσοστό διαθέσιμου χώρου
- I/O - Σας επιτρέπει να παρακολουθείτε τις λειτουργίες ανάγνωσης και εγγραφής των δίσκων.

Παρακολούθηση δικτύου - Τα κύρια στοιχεία δικτύου που είναι ιδανικό να παρακολουθείτε είναι:

- Εισερχόμενη κίνηση
- Εξερχόμενη κίνηση
- Αριθμός σφαλμάτων αποστολής
- Προσαρμογείς δικτύου (Up ή Down)

Παρακολούθηση λειτουργικού συστήματος - Τα κύρια στοιχεία λειτουργικού συστήματος που είναι ιδανικό να παρακολουθείτε είναι:

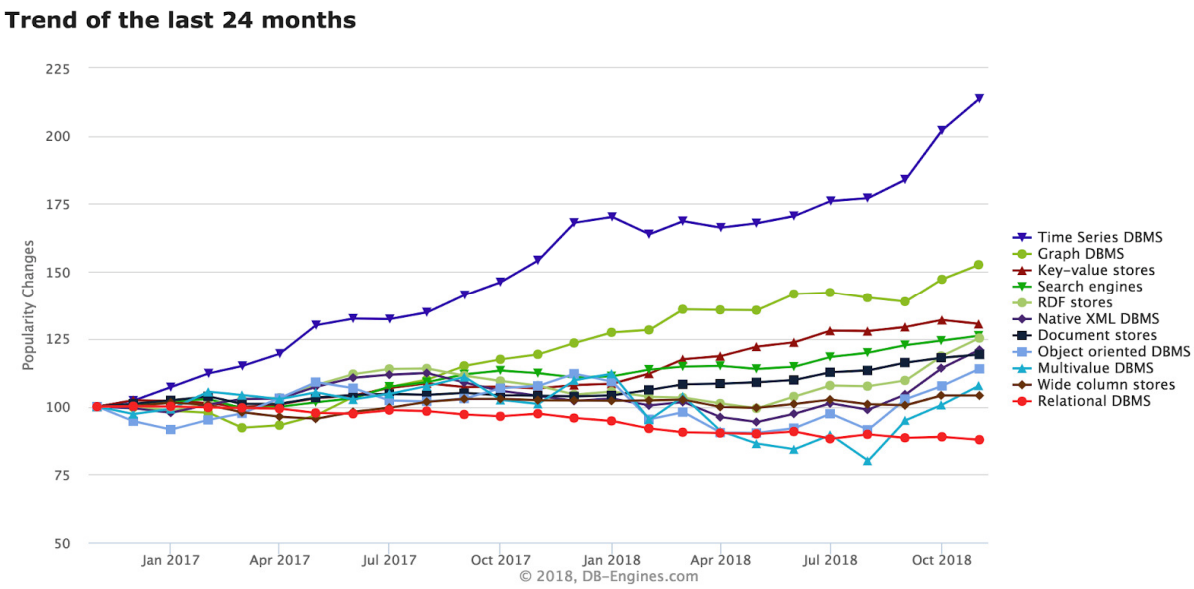
- Uptime
- Πληροφορίες συστήματος
- /etc/passwdChecksum
- Hostname
- Αριθμός συνδεδεμένων χρηστών στο σύστημα

Βάσεις δεδομένων χρονοσειρών

Τι είναι μια βάση δεδομένων χρονοσειρών;

Μια Βάση Δεδομένων Σειράς Χρόνου (TSDB) είναι μια βάση δεδομένων βελτιστοποιημένη για χρονικά σήματα ή δεδομένα χρονοσειρών. Τα δεδομένα χρονολογικών σειρών είναι απλώς μετρήσεις ή συμβάντα που παρακολουθούνται, μειώνονται και συσσωρεύονται με την πάροδο του χρόνου. Αυτά θα μπορούσαν να είναι μετρήσεις διακομιστή, παρακολούθηση απόδοσης εφαρμογών, δεδομένα δικτύου, δεδομένα αισθητήρων, συμβάντα, κλικ, συναλλαγές σε μια αγορά και πολλούς άλλους τύπους δεδομένων ανάλυσης.

Μια βάση δεδομένων χρονοσειρών είναι ειδικά σχεδιασμένη για το χειρισμό μετρήσεων και συμβάντων ή μετρήσεων με χρονοσήμανση. Ένα TSDB είναι βελτιστοποιημένο για τη μέτρηση της μεταβολής με την πάροδο του χρόνου. Οι ιδιότητες που κάνουν τα δεδομένα χρονολογικών σειρών πολύ διαφορετικά από τα άλλα φορτία δεδομένων είναι η διαχείριση του κύκλου ζωής των δεδομένων, η σύνοψη και οι μεγάλες κλίμακες ανίχνευσης πολλών αρχείων.



Γιατί είναι σημαντική η βάση δεδομένων χρονοσειρών;

Οι βάσεις δεδομένων χρονοσειρών δεν είναι καινούργιες, αλλά οι βάσεις δεδομένων της σειράς 1 της πρώτης γενιάς επικεντρώθηκαν κατά κύριο λόγο στην εξέταση των οικονομικών στοιχείων, της μεταβλητότητας των συναλλαγών σε μετοχές και των συστημάτων που δημιουργήθηκαν για την επίλυση των συναλλαγών. Ωστόσο, οι θεμελιώδεις συνθήκες της πληροφορικής έχουν αλλάξει δραματικά την τελευταία δεκαετία. Όλα έχουν κατακερματιστεί. Τα μονολιθικά mainframes έχουν εξαφανιστεί, έχουν αντικατασταθεί από διακομιστές χωρίς μικροεπεξεργαστές και containers.

Επιπλέον, παρακολουθούμε τα όργανα κάθε διαθέσιμης επιφάνειας στους δρόμους, τα αυτοκίνητα, τα εργοστάσια, τα ηλεκτρικά δίκτυα, τους δορυφόρους, τα ρούχα, τα τηλέφωνα, τα μικροκύματα, τους πλανήτες, τα ανθρώπινα σώματα. Όλα έχουν ή θα έχουν αισθητήρα. Έτσι τώρα, όλα μέσα και έξω από την εταιρεία εκπέμπουν μια αδιάκοπη ροή μετρήσεων και γεγονότων ή δεδομένων χρονοσειρών. Αυτό σημαίνει ότι οι υποκείμενες πλατφόρμες πρέπει να εξελιχθούν για να υποστηρίξουν αυτούς τους νέους φόρτους εργασίας - περισσότερα σημεία δεδομένων, περισσότερες πηγές δεδομένων, περισσότερη παρακολούθηση, περισσότερους ελέγχους. Αυτό που παρακολουθούμε και ό, τι απαιτεί η εποχή είναι μια παραδειγματική μετατόπιση του τρόπου με τον οποίο προσεγγίζουμε την υποδομή δεδομένων μας και τον τρόπο

προσέγγισης των συστημάτων οικοδόμησης, παρακολούθησης, ελέγχου και διαχείρισης. Αυτό που χρειαζόμαστε είναι ένα σύγχρονο TSDB.

Οι Βάσεις Δεδομένων έχουν βασικές ιδιότητες αρχιτεκτονικού σχεδιασμού που τις κάνουν πολύ διαφορετικές από άλλες βάσεις δεδομένων. Αυτές περιλαμβάνουν: αποθήκευση και συμπίεση δεδομένων χρονικής σήμανσης, διαχείριση δεδομένων κύκλου ζωής, συνοπτική παρουσίαση δεδομένων, ικανότητα χειρισμού μεγάλων χρονοσειρών εξαρτώμενων σαρώσεων πολλών εγγραφών και ερωτημάτων επίγνωσης χρονοσειρών.

Για παράδειγμα: Με μια βάση δεδομένων χρονοσειρών, είναι κοινό να ζητείται μια σύνοψη δεδομένων για μεγάλο χρονικό διάστημα. Αυτό απαιτεί να μεταβείτε σε ένα εύρος σημείων δεδομένων για να εκτελέσετε ορισμένους υπολογισμούς όπως μια εκατοστιαία αύξηση αυτού του μήνα μετρικής κατά την ίδια περίοδο τους τελευταίους έξι μήνες, που συνοψίζονται κατά μήνα. Αυτό το είδος φόρτου εργασίας είναι πολύ δύσκολο να βελτιστοποιηθεί με ένα κατανεμημένο κατάστημα αξίας κλειδιών. Τα TSDB είναι βελτιστοποιημένα για τη συγκεκριμένη περίπτωση χρήσης, δίνοντας χιλιάδες δευτερόλεπτα σε ερωτηματολογικούς χρόνους σε μήνες δεδομένων. Ένα άλλο παράδειγμα: Με τις Βάσεις Δεδομένων της Σειράς Χρόνου είναι κοινό να διατηρείτε τα δεδομένα υψηλής ακρίβειας γύρω από ένα σύντομο χρονικό διάστημα. Αυτά τα δεδομένα συγκεντρώνονται και υποδεικνύονται σε πιο μακροπρόθεσμα δεδομένα τάσεων. Αυτό σημαίνει ότι για κάθε σημείο δεδομένων που εισέρχεται στη βάση δεδομένων, θα πρέπει να διαγραφεί μετά την πάροδο του χρόνου. Αυτό το είδος διαχείρισης του κύκλου ζωής των δεδομένων είναι δύσκολο για τους προγραμματιστές εφαρμογών να το εφαρμόσουν πάνω από τακτικές βάσεις δεδομένων. Πρέπει να σχεδιάσουν προγράμματα για τη φθηνή έξοδο μεγάλων σειρών δεδομένων και να συνοψίζουν συνεχώς τα δεδομένα αυτά σε κλίμακα. Με μια Βάση Δεδομένων της Σειράς Χρόνου αυτή η λειτουργία παρέχεται εκτός πλαισίου.

Γιατί χρειάζομαι μια βάση δεδομένων χρονοσειρών;

Μπορεί να ρωτήσετε: Γιατί δεν μπορώ να χρησιμοποιήσω απλά μια βάση δεδομένων "κανονική" (δηλ. Μη χρονοσειρών);

Η αλήθεια είναι ότι μπορείτε, και μερικοί άνθρωποι το κάνουν. Ωστόσο, γιατί οι TSDBs είναι η ταχύτερα αναπτυσσόμενη κατηγορία βάσεων δεδομένων σήμερα; Δύο λόγοι: (1) κλίμακα και (2) χρηστικότητα.
Κλίμακα: Τα δεδομένα χρονοσειρών συσσωρεύονται πολύ γρήγορα. (Για παράδειγμα, ένα μόνο συνδεδεμένο αυτοκίνητο θα συλλέγει 4.000 GB δεδομένων ανά ημέρα.) Και οι συνήθεις βάσεις δεδομένων δεν έχουν σχεδιαστεί για να χειρίζονται αυτήν την κλίμακα. Οι σχεσιακές βάσεις δεδομένων τείνουν να είναι ελάχιστα με πολύ μεγάλα σύνολα δεδομένων. Οι βάσεις δεδομένων NoSQL ταιριάζουν καλύτερα στην κλίμακα, αλλά μπορούν ακόμα να ξεπεραστούν από μια βάση δεδομένων που έχει ρυθμιστεί καλά για δεδομένα χρονοσειρών. Αντίθετα, οι βάσεις δεδομένων χρονοσειρών (που μπορούν να βασίζονται σε σχεσιακές ή NoSQL βάσεις δεδομένων) χειρίζονται την κλίμακα εισάγοντας αποτελεσματικότητες που είναι δυνατές μόνο όταν αντιμετωπίζετε το χρόνο ως πολίτης πρώτης τάξης. Αυτές οι βελτιώσεις αποδόσεων έχουν ως αποτέλεσμα βελτιώσεις της απόδοσης, όπως υψηλότερα ποσοστά κατανάλωσης, ταχύτερα ερωτήματα σε κλίμακα (αν και ορισμένα υποστηρίζουν περισσότερα ερωτήματα από άλλα) και καλύτερη συμπίεση δεδομένων.

Ευχρηστία: Οι TSDB περιλαμβάνουν επίσης συνήθεις λειτουργίες και λειτουργίες για την ανάλυση δεδομένων χρονοσειρών όπως πολιτικές διατήρησης δεδομένων, συνεχείς ερωτήσεις, ευέλικτες συνθέσεις χρόνου κλπ. Ακόμη και αν η κλίμακα δεν ανησυχεί αυτή τη στιγμή (π.χ. συλλογή δεδομένων), αυτά τα χαρακτηριστικά μπορούν να παρέχουν ακόμα καλύτερη εμπειρία χρήστη και να διευκολύνουν τη ζωή σας.

Αυτός είναι ο λόγος για τον οποίο οι προγραμματιστές υιοθετούν όλο και περισσότερο βάσεις δεδομένων χρονολογικών σειρών και τις χρησιμοποιούν για διάφορες περιπτώσεις χρήσης:

- Παρακολούθηση συστημάτων λογισμικού: Εικονικές μηχανές, containers, υπηρεσίες, εφαρμογές
- Παρακολούθηση των φυσικών συστημάτων: Εξοπλισμός, μηχανήματα, συνδεδεμένες συσκευές, περιβάλλον, σπίτια μας, σώμα μας

- Εφαρμογές παρακολούθησης περιουσιακών στοιχείων: Οχήματα, φορτηγά, φυσικά εμπορευματοκιβώτια, παλέτες
- Συστήματα χρηματοοικονομικών συναλλαγών: Κλασσικά χρεόγραφα, νεότερα κρυπτοσυχνότητες
- Εφαρμογές συμβάντων: Παρακολούθηση δεδομένων αλληλεπίδρασης χρήστη / πελάτη

Εργαλεία επιχειρηματικής ευφυΐας: Παρακολούθηση των βασικών μετρήσεων και της συνολικής υγείας της επιχείρησης, και άλλα.

Ακόμα και τότε, θα χρειαστεί να επιλέξετε μια βάση δεδομένων χρονικής σειράς που να ταιριάζει καλύτερα στο μοντέλο δεδομένων σας και να γράψετε / διαβάσετε σχέδια.

Zabbix

ToZabbix, δημιουργήθηκε απο το τον AlexeiVladishev, και αναπτύσσεται και υποστηρίζεται από την Zabbix SIA.

ToZabbix είναι ανοιχτού κώδικα εργαλείο παρακολούθησης συστημάτων και προγραμμάτων, για επιχειρήσεις και μη. Είναι ένα πρόγραμμα που παρακολουθεί μεγάλο αριθμό παραμέτρων ενός δικτύου και την κατάσταση και την ακεραιότητα των servers. Μπορεί να παρακολουθήσει λειτουργίες σε Linux, HewlettPackardUnix (HP-UX), Mac OS X, Solaris και άλλα λειτουργικά συστήματα. Ωστόσο, η παρακολούθηση των Windows είναι δυνατή μόνο μέσω zabbixagent. Έχει τη δυνατότητα να χρησιμοποιήσει βάσεις δεδομένων όπως MySQL, MariaDB, PostgreSQL, SQLite, Oracle ή IBM DB2 για την αποθήκευση δεδομένων. Το backend είναι γραμμένο σε C και το frontend είναι γραμμένο σε PHP. Το Zabbix προσφέρει πολλές επιλογές παρακολούθησης. Χρησιμοποιεί ένα βολικό μηχανισμό ειδοποιήσεων, που επιτρέπει στους χρήστες να ρυθμίζουν τις ειδοποιήσεις με βάση το ηλεκτρονικό ταχυδρομείο για σχεδόν οποιοδήποτε γεγονός. Αυτό επιτρέπει στους χρήστες να αντιδράσουν γρήγορα σε προβλήματα που μπορεί να προκληθούν στους servers. Παράγει εξαιρετικές αναφορές και πολλά χαρακτηριστικά απεικόνισης βασισμένα στα αποθηκευμένα δεδομένα. Αυτό κάνει το Zabbix ιδανικό για πρόβλεψη χωρητικότητας.

Το zabbix υποστηρίζει polling και trapping. Topolling γίνεται όταν ο serverανα τακτά χρονικά διαστήματα ρωτάει τους agents για να πάρει δεδομένα. Το trapping από την άλλη γίνεται, όταν οι agents από μόνοι τους στέλνουν δεδομένα στο server. Όλες οι αναφορές και στατιστικά

στοιχεία του Zabbix, καθώς και οι παράμετροι διαμόρφωσης, μπορούν ανακτηθούν μέσω ενός διαχειριστικού site.

Το διαχειριστικό site διασφαλίζει ότι η κατάσταση του δικτύου σας και η κατάσταση των servers σας μπορούν να εκτιμηθούν από οποιαδήποτε τοποθεσία. Το Zabbix είναι κατάλληλα διαμορφωμένο έτσι ώστε μπορεί να παίξει σημαντικό ρόλο στην παρακολούθηση μιας υποδομής πληροφορικής.. Αυτό ισχύει εξίσου για μικρούς οργανισμούς με λίγους servers και για μεγάλες εταιρείες με μεγάλο αριθμό servers.

Το Zabbix είναι δωρεάν. Γράφεται και διανέμεται υπό την έκδοση της άδειας του GPL 2 . Σημαίνει ότι ο πηγαίος κώδικας του είναι ελεύθερα διανεμημένος και διατίθεται για το ευρύ κοινό.

Αρχιτεκτονική του Zabbix

Το Zabbix αποτελείται από πολλά σημαντικά συστατικά λογισμικού:

Server

Ο Zabbixserver είναι το κεντρικό σημείο στο οποίο οι zabbixagents στέλνουν δεδομένα σχετικά με τη διαθεσιμότητα και την ακεραιότητα των συστημάτων. Ο server είναι ο κεντρικός χώρος αποθήκευσης στον οποίο αποθηκεύονται όλες οι ρυθμίσεις και δεδομένα στατιστικής και λειτουργίας.

Database

Όλες οι ρυθμίσεις καθώς και τα δεδομένα που συλλέγονται από τον Zabbix αποθηκεύονται σε μια βάση δεδομένων.

Web Interface

Για εύκολη πρόσβαση στο Zabbix από οπουδήποτε και από οποιαδήποτε πλατφόρμα παρέχεται η διαδικτυακή διασύνδεση. Το webinterface είναι μέρος του Zabbixserver και συνήθως τρέχει στο ίδιο μηχάνημα με αυτό που τρέχει και ο Zabbixserver.

Proxy

Ο Zabbixproxy μπορεί να συλλέγει δεδομένα σχετικά με την απόδοση και διαθεσιμότητα για τον Zabbixserver. Ένας proxy είναι προαιρετικό μέρος για την ανάπτυξη μιας υποδομής

παρακολούθησης με Zabbix. Τον χρησιμοποιούν κυρίως σε κρίσιμες υποδομές εγκαθιστώντας τον στο ίδιο datacenter ή ακόμα και στο ίδιο μηχάνημα με αυτό που τρέχουν οι κρίσιμες υπηρεσίες και βρίσκονται πίσω από κάποιο firewall. Αυτός με τη σειρά του επικοινωνεί με τον zabbixserver και του στέλνει όλα τα δεδομένα όποτε μπορεί, π.χ. σε ένα δικτυακό πρόβλημα που μπορεί να προκύψει, ο proxy θα συλλέξει τα δεδομένα από τα συστήματα και τις δικτυακές συσκευές και θα στείλει τα δεδομένα όταν διορθωθεί το πρόβλημα. Ωστόσο, μπορεί να είναι πολύ ωφέλιμο με σκοπό να διανέμεται το φορτίο ενός μόνο διακομιστήZabbix.

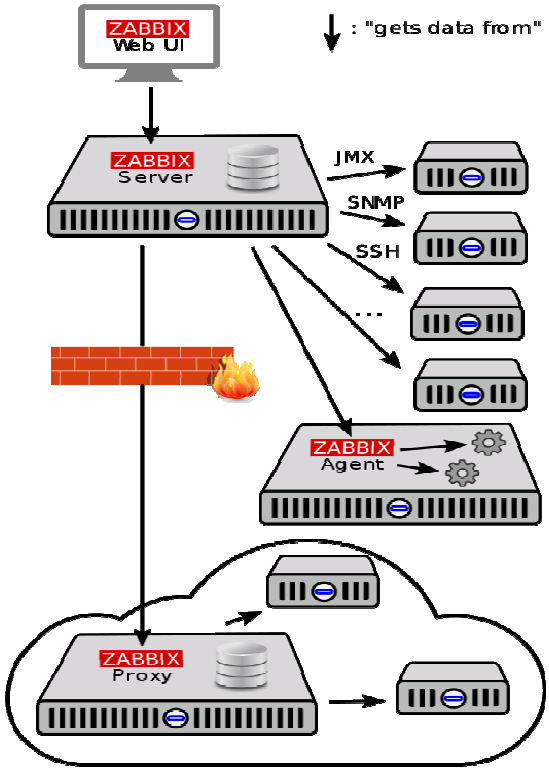
Zabbixagent

Οι Zabbixagents εγκαθίστανται στους στόχους παρακολούθησης για να παρακολουθούν ενεργά τους τοπικούς πόρους και εφαρμογές και να στέλνουν τα δεδομένα που έχουν συγκεντρώσει στο Zabbixserver.

DataFlow

Επιπλέον, είναι σημαντικό να δούμε τη συνολική ροή δεδομένων στο Zabbix. Για να δημιουργήσετε ένα στοιχείο που συγκεντρώνει δεδομένα, πρέπει πρώτα να δημιουργήσετε έναν zabbixagent. Προχωρώντας στο άλλο άκρο του φάσματος Zabbix, πρέπει πρώτα να έχετε ένα zabbixitem για να δημιουργήσετε trigger. Πρέπει να έχετε μια trigger για να δημιουργήσετε μια ενέργεια. Έτσι, αν θέλετε να λάβετε μια ειδοποίηση ότι η χρήση του CPU σας έχει ανέβει αρκετά στον X μηχάνημα, πρέπει πρώτα να δημιουργήσετε το item για το X μηχάνημα για την παρακολούθηση της χρήσης CPU του και με αυτό τον τρόπο ενεργοποιείται το trigger. Αν η χρήση του CPU έχει ανέβει αρκετά, με μια ενέργεια στέλνεται ένα μήνυμα ηλεκτρονικού ταχυδρομείου το οποίο αναφέρει το πρόβλημα. Αυτό μπορεί να φαίνεται αρκετά δύσκολο, όμως με τη χρήση του templating είναι εύκολο. Ωστόσο, λόγω αυτού του σχεδιασμού είναι δυνατό να δημιουργηθεί μια εύκολη εγκατάσταση.

Στην παρακάτω εικόνα φαίνεται μια αρχιτεκτονική υποδομής παρακολούθησης με τη χρήση του Zabbix.



Zabbix Server

Ο zabbixserver είναι η κεντρική λειτουργία του λογισμικού zabbix. Ο server εκτελεί τεχνικές συλλογής δεδομένων trapping και polling, υπολογίζει τα triggers και στέλνει ειδοποιήσεις στους χρήστες. Είναι το κεντρικό στοιχείο στο οποίο agents και οι πληρεξούσιοι του zabbixserver αναφέρουν δεδομένα σχετικά με τη διαθεσιμότητα και ακεραιότητα των συστημάτων. Ο ίδιος ο server μπορεί να ελέγξει από μόνος του, εξ αποστάσεως συστήματα όπως webservers και mailservers χρησιμοποιώντας απλούς ελέγχους υπηρεσιών. Ο server είναι επίσης το κεντρικό αποθετήριο στο οποίο αποθηκεύονται όλες οι ρυθμίσεις, τα στατιστικά, τα λειτουργικά δεδομένα και είναι ο φορέας που θα ειδοποιεί ενεργά τους διαχειριστές όταν προκύψουν προβλήματα σε οποιοδήποτε από τα παρακολουθούμενα συστήματα. Η λειτουργία ενός βασικού zabbixserver χωρίζεται σε τρία διαφορετικά στοιχεία, το server, το front-end, και την αποθήκευση στη βάση δεδομένων. Όλες οι πληροφορίες για το zabbix, αποθηκεύονται στη βάση δεδομένων στην οποία

αλληλεπιδρούν τόσο ο server όσο και το γραφικό περιβάλλον. Για παράδειγμα, όταν δημιουργείται ένα νέο item χρησιμοποιώντας το γραφικό περιβάλλον ή το API, προστίθεται στον πίνακα των items στη βάση δεδομένων και, στη συνέχεια, περίπου μία φορά το λεπτό, ο server θα ρωτήσει τον πίνακα για μία λίστα από items που είναι ενεργά, και τα οποία στη συνέχεια αποθηκεύονται σε μία προσωρινή μνήμη στο zabbixserver. Αυτός είναι και ο λόγος για τον οποίο μπορεί να χρειαστούν έως και 2 λεπτά για να εμφανιστούν οι αλλαγές στο γραφικό περιβάλλον του zabbix στο τμήμα του "LatestData".

ZabbixAgent

Ο zabbixagent αναπτύσσεται σε έναν στόχο παρακολούθησης για να παρακολουθεί ενεργά τους τοπικούς πόρους και εφαρμογές ενός υπολογιστή π.χ. (σκληρούς δίσκους μνήμη, επεξεργαστή, στατιστικά και κατάσταση εφαρμογών.

Συγκεντρώνει τα δεδομένα τοπικά και τα στέλνει στον zabbixserver για περαιτέρω επεξεργασία. Σε περίπτωση βλάβης π.χ. (πρόβλημα σε σκληρό δίσκο), ο zabbixserver ειδοποιεί ενεργά τους διαχειριστές του συγκεκριμένου συστήματος που ανέφερε αποτυχία. Είναι εξαιρετικά αποδοτικός λόγω της άμεσης επαφής με το σύστημα (systemcalls) για τη συλλογή στατιστικών πληροφοριών.

Οι zabbixagents μπορούν να εκτελέσουν παθητικούς και ενεργούς ελέγχους. Στον παθητικό έλεγχο zabbixagent ανταποκρίνεται στα ερωτήματα δεδομένων. Ο zabbixserver ρωτάει για δεδομένα όπως cpuload, μνήμη, I/O, και ο zabbixagent του στέλνει τα αποτελέσματα.

Οι ενεργοί έλεγχοι απαιτούν πιο περίπλοκη επεξεργασία. Ο agent πρώτα πρέπει να ανακτήσει μια λίστα με τα items από zabbixserver για ανεξάρτητη επεξεργασία και τότε ανά τακτά χρονικά διαστήματα, στέλνει τις νέες τιμές στο server.

Υποστηριζόμενες πλατφόρμες:

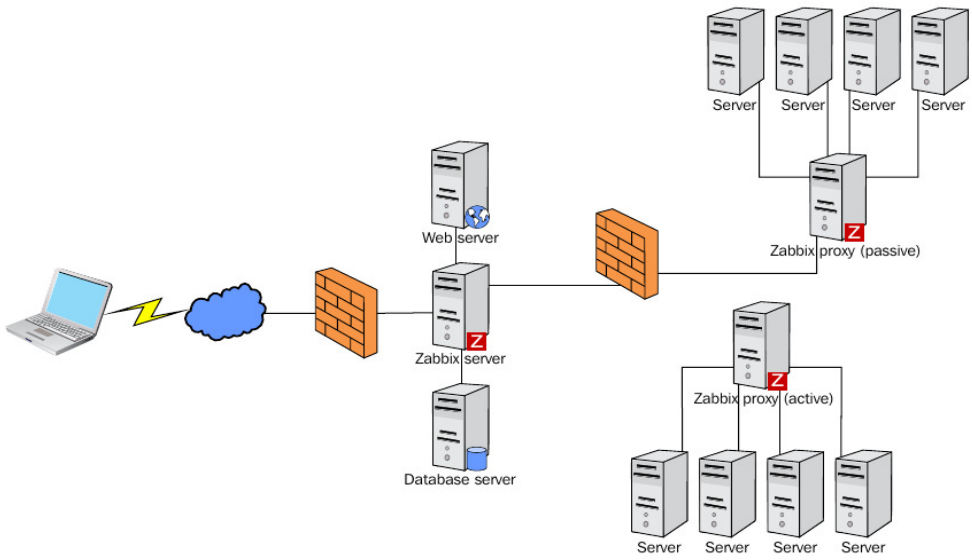
- Linux
- IBM AIX
- FreeBSD
- NetBSD
- OpenBSD

- HP-UX
- Mac OS X
- Solaris: 8, 9, 10, 11
- Windows

Zabbix Proxy

Ο zabbixproxy είναι ένα ακόμα μέλος της σουίτας προγραμμάτων του zabbix το οποίο κάθεται ανάμεσα από τον κεντρικό zabbixserver και το zabbixagent. Όπως και ο server χρησιμοποιείται για τη συλλογή δεδομένων από στοιχεία (items) μηχανημάτων (hosts) και μπορεί να διατηρεί τα δεδομένα για μία αυθαίρετη χρονική στιγμή, αναμεταδίδοντας τα σε μια βάση δεδομένων, έτσι και ο agent δεν έχει κάποιο γραφικό περιβάλλον και διοικείται απευθείας από τον κεντρικό server. Περιορίζεται επίσης στη συλλογή δεδομένων χωρίς κάποια ενέργεια αξιολόγησης ή κάποιο trigger. Όλα αυτά τα χαρακτηριστικά κάνουν το zabbixproxy ένα απλό και ελαφρύ εργαλείο που αναπτύσσεται αν κάποιος χρειάζεται απομακρυσμένη παρακολούθηση από κάποιο κεντρικό server ή αν ο στόχος είναι ο έλεγχος και η βελτιστοποίηση της ροής των δεδομένων που παρακολουθούνται μέσω των δικτύων, ή και τα δύο.

Στην παρακάτω εικόνα φαίνεται μια βασική κατανεμημένη αρχιτεκτονική συμπεριλαμβάνοντας ZabbixProxies.



Από τη φύση του, ένας zabbixproxy καλό είναι να τρέχει σε ένα αποκλειστικό μηχάνημα το οποίο είναι διαφορετικό από τον κύριο server. Ένας proxy αφορά τη συλλογή δεδομένων. Το χαρακτηριστικό του δεν είναι το γραφικό περιβάλλον και η βελτίωση κάποιων περίπλοκων ερωτημάτων ή υπολογισμών. Επομένως, δεν είναι απαραίτητο να αναθέσουμε ένα σχετικά δυνατό μηχάνημα με πολύ επεξεργαστική ισχύ ή μεγάλο throughput δίσκου. Στην πραγματικότητα μία σχετικά φτωχή υποδομή υλικού είναι συχνά καλύτερη επιλογή. Τα μηχανήματα που τρέχουν οι zabbixproxies πρέπει είναι αρκετά ελαφριά όχι μόνο για την απλότητα του λογισμικού αλλά επίσης γιατί πρέπει να είναι εύκολη και χαμηλού κόστους παραμετροποιήσιμη επιλογή για να επεκτείνεις και να κατανέμεις την αρχιτεκτονική του monitoringχωρίς να δημιουργήσεις πάρα πολλές επιπτώσεις στην ανάπτυξη και το κόστος.

Είναι πιθανή η εξαίρεση της μικρής φθηνής και λιτής κατευθυντήριας γραμμής για zabbixproxies που μπορούν να ανακύψουν αν καταλήξεις στην ανάθεση χιλιάδων items χιλιάδων μηχανημάτων σε ένα proxy. Σε αυτή την περίπτωση αντί να αναβαθμίσεις το υλικό του μηχανήματος σε πιο δυνατό μηχάνημα, είναι συχνά φθηνότερο να χωρίσεις τους host σε διαφορετικά groups και να τους αναθέσεις σε διαφορετικούς μικρότερους proxies. Στις περισσότερες περιπτώσεις, αυτή θα ήταν η προτιμώμενη επιλογή καθώς το φορτίο δε διανέμεται και εξισορροπείται απλώς, αλλά εξετάζεται επίσης η πιθανότητα τεράστιας απώλειας δεδομένων, εάν ένα μόνο μηχάνημα επιφορτισμένο με την παρακολούθηση ενός μεγάλου μέρους του δικτύου έπεφτε για οποιονδήποτε λόγο.

Εξετάζοντας τη χρήση μικρών, ελαφρών μηχανημάτων ως zabbixproxies, τείνουν να είναι φθηνά, εύκολο να τα αναπτύξεις, αξιόπιστα και αρκετά οικονομικά όταν πρόκειται για απαιτήσεις ισχύος. Έχουν ιδανικά χαρακτηριστικά για οποιαδήποτε λύση παρακολούθησης που έχει ως στόχο να παρακολουθεί όσο το δυνατόν περισσότερο το σύστημα. Υπάρχει μια άλλη εκδοχή: αν έχετε ένα διαχωρισμένο δίκτυο, το οποίο ίσως μοιράζεται ακόμη και σε πολλές διαφορετικές γεωγραφικές τοποθεσίες, είναι προτιμότερο να εξεταστεί μια πολύ καλή σταθερή βάση δεδομένων πίσω από αυτό. Αυτός ο λόγος οφείλεται στο γεγονός ότι μια διακοπή δικτύου, που μπορεί να διαρκέσει για ένα μεγάλο χρονικό διάστημα, θα αναγκάσει τον proxy να διατηρήσει μια σημαντική ποσότητα δεδομένων για μια μεγάλη χρονική περίοδο, και εδώ, αν ο proxy πέσει, αυτό μπορεί να είναι ένα σοβαρό πρόβλημα.

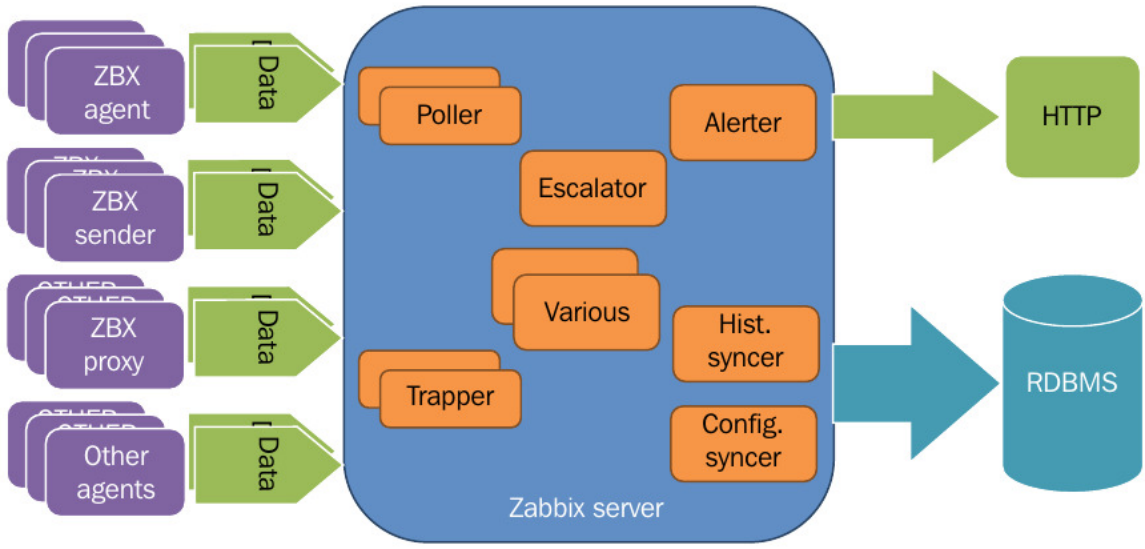
Όπως είπαμε, η ποσοτικοποίηση της χρονικής περιόδου που χρειάζεται ο proxy για να επιβιώσει χωρίς σύνδεση με τον κεντρικό server, μπορεί να είναι αρκετά περίπλοκη, καθώς εξαρτάται από δύο συγκεκριμένους παράγοντες: τον αριθμό των κεντρικών υπολογιστών που παρακολουθούνται από συγκεκριμένο proxy και επιπλέον, ο αριθμός των αντικειμένων ή των μετρήσεων που έχει αποκτήσει ο proxy πρέπει να αποθηκευτεί στην τοπική βάση δεδομένων. Εδώ, είναι εύκολο να καταλάβουμε ότι αυτή η εκδοχή, θα οδηγήσει στην επιλογή της βάσης δεδομένων. Εάν ο proxy βρίσκεται στο τοπικό σας δίκτυο ή όχι, η απόφαση θα είναι υπέρ μιας ελαφριάς και αποδοτικής βάσης δεδομένων, όπως το SQLite3. Διαφορετικά, θα είμαστε υποχρεωμένοι να επιλέξουμε ένα διαφορετικό είδος βάσης δεδομένων που μπορεί να διατηρεί δεδομένα για μεγάλο χρονικό διάστημα και μπορεί να είναι πιο ανεκτική σε σύγκρουση από την MySQL ή PostgreSQL.

ZabbixDataFlow

Είναι σημαντικό να έχετε τουλάχιστον μια ιδέα της τυπικής ροής δεδομένων παρακολούθησης του Zabbix. Μπορούμε να έχουμε τουλάχιστον τέσσερα διαφορετικά είδη πηγών δεδομένων που μπορούν να παραδώσουν στοιχεία στο Zabbix Server:

- Ο Zabbixagent
- Η εντολή zabbix_send του αποστολέα Zabbix
- Τρίτοι οι οποίοι μπορούν να στείλουν με τη βοήθεια του API
- Zabbix Proxy

Το παρακάτω διάγραμμα απεικονίζει την απλοποιημένη ροή δεδομένων που ακολουθείται ένα στοιχείο Zabbix:



Zabbixitems& Ροή δεδομένων

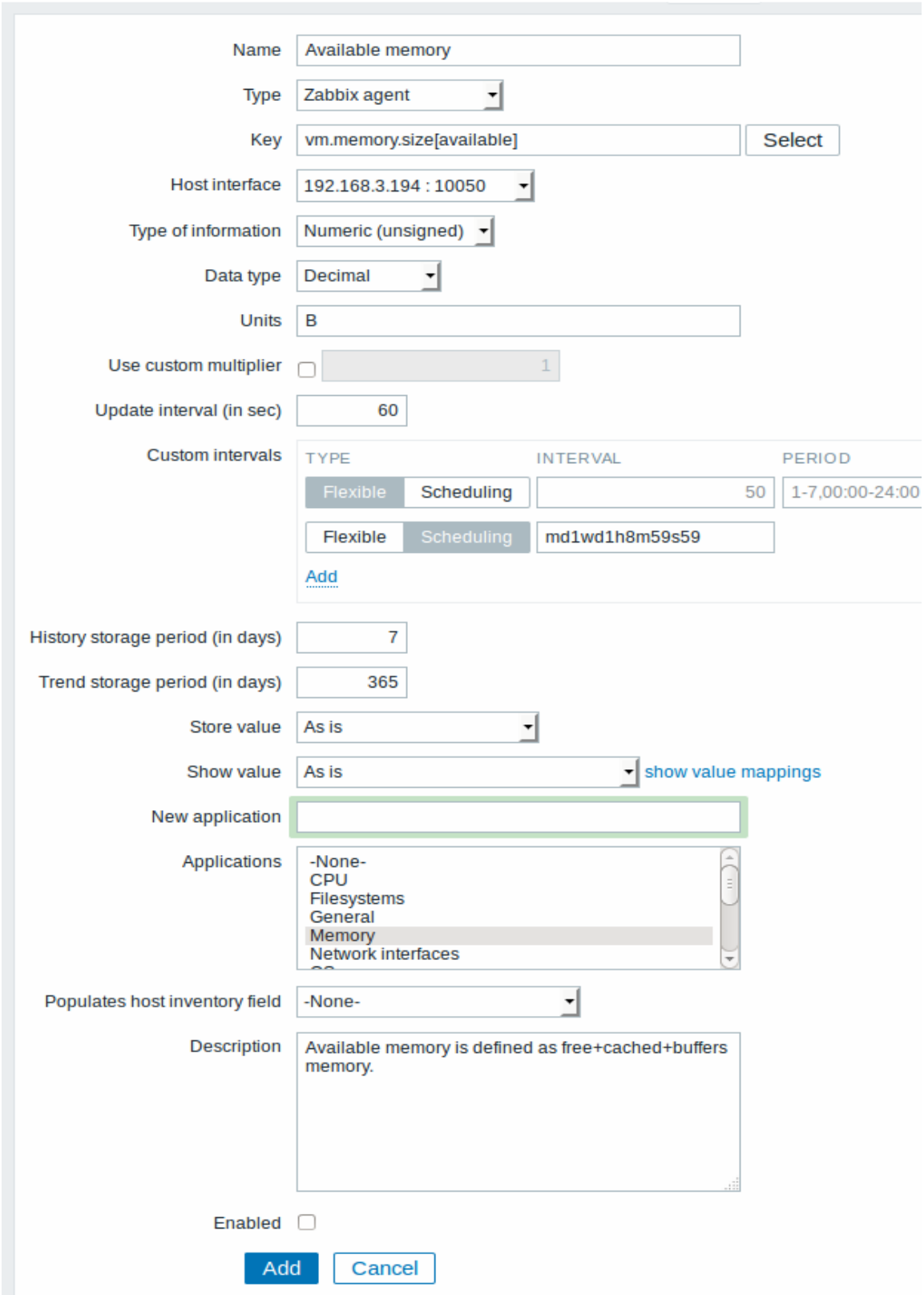
Items

Τα items, είναι αυτά που συλλέγουν δεδομένα από έναν host. Μόλις προσθέσετε έναν host στον zabbixserver, πρέπει να προσθέσετε ορισμένα items για να λάβετε τα δεδομένα .

Ένα item είναι μια μεμονωμένη μέτρηση. Ένας τρόπος για να προσθέσετε γρήγορα πολλά items, είναι να επισυνάψετε ένα από τα προκαθορισμένα templates σε έναν host. Ωστόσο, για βελτιστοποιημένη απόδοση του συστήματος, ίσως χρειαστεί να προσαρμόσετε τα templates ώστε να έχετε τόσα items, και να τα παρακολουθείτε τόσο συχνά, όσο είναι απαραίτητο.

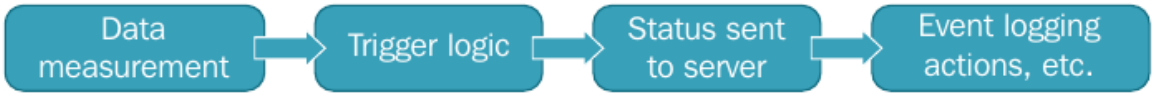
Σε ένα ξεχωριστό item καθορίζετε τι είδους δεδομένα θα συλλέγονται από τον host. Για το σκοπό αυτό, χρησιμοποιήστε το κλειδί (key) του item. Έτσι, ένα item με το όνομα system.cpu.load θα συλλέξει δεδομένα του cpuload, ενώ ένα item με το όνομα κλειδιού net.if. θα συλλέξει δεδομένα εισερχόμενης κίνησης δικτύου. Για να ορίσετε πρόσθετες παραμέτρους με το κλειδί, πρέπει συμπεριλάβετε σε αγκύλες μετά το όνομα του κλειδιού. Έτσι, το system.cpu.load [avg5] θα επιστρέψει το μέσο cpuload για τα τελευταία 5 λεπτά, ενώ το net.if.in [eth0] θα εμφανίσει την εισερχόμενη κίνηση για το eth0 προσαρμογέα δικτύου.

Για να δημιουργήσετε ένα zabbixitem στο γραφικό περιβάλλον του zabbix χρειάζεται να συμπληρώσετε κάποια από τα attributes όπως φαίνεται στην παρακάτω εικόνα.

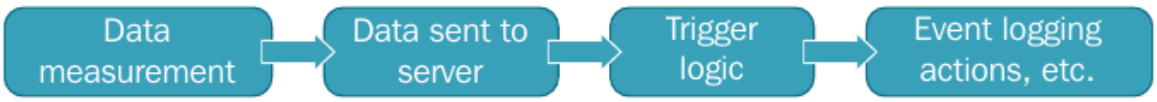

Name
Available memory
Type
Zabbix agent
Key
vm.memory.size[available]
Select
Host interface
192.168.3.194 : 10050
Type of information
Numeric (unsigned)
Data type
Decimal
Units
B
Use custom multiplier
1
Update interval (in sec)
60
Custom intervals
TYPE
INTERVAL
PERIOD
Flexible
Scheduling
50
1-7,00:00-24:00
Flexible
Scheduling
md1wd1h8m59s59
Add
History storage period (in days)
7
Trend storage period (in days)
365
Store value
As is
Show value
As is
show value mappings
New application
Applications
-None-
CPU
Filesystems
General
Memory
Network interfaces
Populates host inventory field
-None-
Description
Available memory is defined as free+cached+buffers memory.
Enabled
Add
Cancel

Συγκέντρωση zabbixitems ως ακατέργαστα δεδομένα

Ένα από τα πιο σημαντικά χαρακτηριστικά που έχει το Zabbix σε σχέση με τις περισσότερες λύσεις παρακολούθησης συστημάτων είναι ότι η κύρια λειτουργία με την οποία αλληλεπιδρά με τα αντικείμενα που παρακολουθούνται εστιάζει στη συλλογή ανεπεξέργαστων δεδομένων, σε αντίθεση με ειδοποιήσεις ή ενημερώσεις κατάστασης. Με άλλα λόγια, πολλά εργαλεία παρακολούθησης εφαρμόζουν τη ροή δεδομένων, όπως φαίνεται στο το ακόλουθο διάγραμμα:



Δηλαδή, ένας zabbixagent ή οποιοσδήποτε άλλος ανιχνευτής παρακολούθησης καλείται να λάβει όχι μόνο τη μέτρηση, αλλά και να ενσωματώσει ένα είδος απόφασης σχετικά με την κατάσταση του, πριν την αποστείλει στο στοιχείο του κύριου server για περαιτέρω επεξεργασία. Από την άλλη πλευρά, η βασική ροή εργασίας του Zabbix είναι ελαφριά, διαφορετική, όπως φαίνεται στο παρακάτω διάγραμμα:



Εδώ, ένας agent ή ένας ανιχνευτής παρακολούθησης δημιουργεί τη μέτρηση και στη συνέχεια στέλνει την εν λόγω μέτρηση στο στοιχείο του server για αποθήκευση και ενδεχομένως για περαιτέρω επεξεργασία. Τα δεδομένα δεν σχετίζονται με συγκεκριμένη απόφαση ενεργοποίησης (pass / fail, ok / προειδοποίηση /σφάλμα ή οποιαδήποτε άλλη παραλλαγή) αλλά διατηρούνται στον server ως ένα μοναδικό σημείο δεδομένων ή μέτρηση. Όπου ισχύει, δηλαδή, για τους αριθμητικούς τύπους, διατηρούνται επίσης σε τύπους δεδομένων ως ελάχιστο, μέγιστο και μέση τιμή, έναντι διαφορετικών χρονικών περιόδων. Διατηρώντας τα δεδομένα χωριστά από τη λογική αποφάσεων, αλλά ένα μόνο σημείο (μια βάση δεδομένων), δίνει στο Zabbix δύο ξεχωριστά πλεονεκτήματα.

Το πρώτο είναι ότι μπορείτε να χρησιμοποιήσετε το Zabbix για να συλλέξετε δεδομένα σχετικά με πράγματα που δεν είναι άμεσα σχετικά με τις πιθανές ειδοποιήσεις και ενέργειες που πρέπει

να λάβετε, αλλά σχετίζονται με το συνολική απόδοση και συμπεριφορά ενός συστήματος. Ένα κλασικό παράδειγμα είναι αυτό ενός switch με πολλές θύρες. Μπορεί να μην ενδιαφέρεστε να ενημερωθείτε για την ανώμαλη κίνηση σε κάθε θύρα, αλλά μπορεί να σας ενδιαφέρει η συγκέντρωση τόσο της μέτρησης της κίνησης σε επίπεδο θύρας όσο και σε επίπεδο μεταγωγής, προκειμένου να καθοριστεί μια γραμμή βάσης, για να αξιολογήσετε πιθανά σημεία συμφόρησης ή να σχεδιάσετε μια επέκταση της υποδομής δικτύου σας.

Παρόμοιες περιπτώσεις μπορούν να γίνουν σχετικά με την CPU και τη χρήση πυρήνα, την ικανότητα αποθήκευσης, τον αριθμό των ταυτόχρονων χρηστών σε μια δεδομένη εφαρμογή και πολλά άλλα. Στην απλούστερη του περίπτωση, ο Zabbix θα μπορούσε να χρησιμοποιηθεί ακόμη και για να συγκεντρώσει τα δεδομένα χρήσης και να το απεικονίσει σε διαφορετικά γραφήματα, χωρίς καν να χρησιμοποιήσετε τα ισχυρά χαρακτηριστικά των triggers και συσχετισμού και να αποδειχθεί ακόμα να είναι μια εξαιρετική επένδυση του χρόνου και των πόρων σας.
Το δεύτερο μεγάλο πλεονέκτημα της κατοχής μιας πλήρους, κεντρικής βάσης δεδομένων ανεπεξέργαστων δεδομένων σε αντίθεση με μια μόνο μέτρηση είναι ότι, για κάθε λογική ενός trigger και μιας απόφασης, που πρέπει να πάρεις, μπορείτε να χρησιμοποιήσετε ολόκληρη τη βάση δεδομένων μέτρησης για να ορίσετε ακριβώς το είδος του γεγονότος που θέλετε να παρακολουθήσετε και να ειδοποιηθείτε. Δεν χρειάζεται να βασίζεστε σε μία μόνο μέτρηση. Δεν χρειάζεται καν να βασιστείτε στις τελευταίες μετρήσεις, συν μερικές από τις προηγούμενες από το ίδιο αντικείμενο ή να περιορίσετε τον εαυτό σας σε στοιχεία από το ίδιο μηχάνημα. Στην πραγματικότητα, μπορείτε να συσχετίσετε οτιδήποτε με οτιδήποτε άλλο στο ιστορικό στοιχείων στη βάση δεδομένων. Θα αρκούσε να πούμε ότι όλη αυτή η δύναμη βασίζεται στο γεγονός ότι το Zabbix διαχωρίζει εντελώς τις λειτουργίες συλλογής δεδομένων από τις λειτουργίες ενεργοποίησης λογικής και δράσης. Όλο αυτό βασίζεται στο γεγονός ότι οι μετρήσεις είναι μόνο μετρήσεις και τίποτα άλλο. Έτσι, στο Zabbix, ένα στοιχείο αντιπροσωπεύει μια μετρική - μία και μόνη πηγή δεδομένων και μετρήσεις.

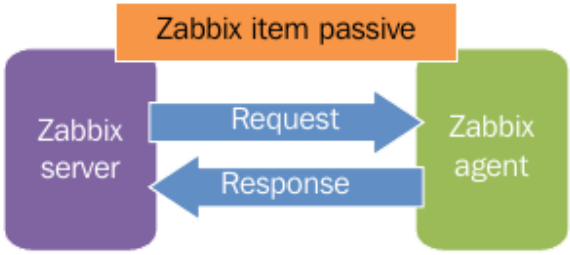
Ροή δεδομένων των Zabbixitems (ενεργών και παθητικών)

Ένα στοιχείο Zabbix μπορεί να γίνει κατανοητό από τα βασικά του στοιχεία - ένα αναγνωριστικό, έναν τύπο δεδομένων και ένας συσχετιζόμενος υπολογιστής. Αυτά είναι τα

στοιχεία που είναι γενικά χρήσιμα για τα υπόλοιπα συστατικά στοιχεία του Zabbix. Το αναγνωριστικό και ο αντίστοιχος υπολογιστής χρησιμοποιούνται για να διακρίνουν ένα στοιχείο μεταξύ των χιλιάδων που μπορούν να οριστούν σε ένα περιβάλλον παρακολούθησης. Ο τύπος δεδομένων είναι σημαντικός, ώστε το Zabbix να γνωρίζει πώς να αποθηκεύει τα δεδομένα, πώς να τα απεικονίζει, και το σημαντικότερο, τι είδους λειτουργίες μπορούν να εφαρμοστούν σε αυτό για να μοντελοποιηθούν οι διεγέρτες και η διαδικασία περαιτέρω.

Το όνομα του στοιχείου είναι μια περιγραφική ετικέτα που προορίζεται να είναι αναγνώσιμη, ενώ το κλειδί του στοιχείου ακολουθεί μια συγκεκριμένη σύνταξη και ορίζει ακριβώς τη μέτρηση που θέλουμε να μετρήσουμε. Δύο άλλα πολύ σημαντικά στοιχεία που είναι κοινά για όλα τα αντικείμενα είναι το ιστορικό και ο τύπος στοιχείου.

Πρώτα απ 'όλα, οι ενεργητικές και παθητικές έννοιες πρέπει να κατανοηθούν από την πλευρά του zabbixagent και όχι του server. Επιπλέον, χρησιμεύουν για την απεικόνιση του που εκκινεί μια σύνδεση για να στείλει ή να λάβει τις πληροφορίες και τα δεδομένα παρακολούθησης, όπως φαίνεται στην παρακάτω εικόνα:

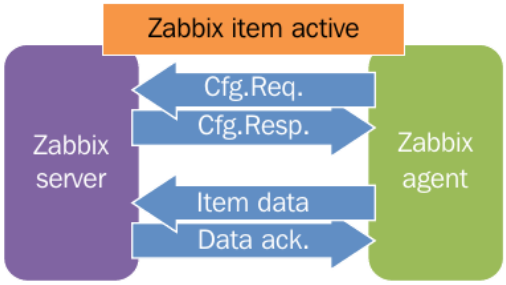


Επομένως, ένα τυπικό στοιχείο Zabbix θεωρείται παθητικό από την άποψη του zabbixagent. Αυτό σημαίνει ότι είναι δουλειά του server να ζητήσει από τον agent, κατά τα χρονικά διαστήματα που ορίζονται, να λάβει την επιθυμητή μέτρηση και να την αναφέρει αμέσως. Όσον αφορά τις λειτουργίες δικτύου, μια μοναδική σύνδεση ξεκινάει από τον server ενώ ο agent βρίσκεται στη λειτουργία ακρόασης. Στο άλλο άκρο, στην περίπτωση ενός ενεργού στοιχείου Zabbix, η δουλειά του agent είναι να ρωτήσει τον server ποια δεδομένα παρακολούθησης πρέπει να συλλέξει και σε ποια χρονικά διαστήματα. Στη συνέχεια προχωράει να προγραμματίσει τις δικές του μετρήσεις και να συνδεθεί πίσω στο server για να τις στείλει για περαιτέρω

επεξεργασία. Όσον αφορά τις λειτουργίες δικτύου, γίνονται δύο ξεχωριστές συνεδρίες που εμπλέκονται στη διαδικασία:

- Ο agent ρωτάει τον server σχετικά με τα στοιχεία και τα χρονικά διαστήματα παρακολούθησης
- Ο agent στέλνει στονserver τα δεδομένα παρακολούθησης που συλλέγει.

Σε αντίθεση με τα τυπικά παθητικά στοιχεία, θα πρέπει να διαμορφωθεί ένας agent έτσι ώστε να γνωρίζει σε ποιον server θα πρέπει να συνδεθεί για λόγους διαμόρφωσης και ανταλλαγής δεδομένων. Αυτό, βεβαίως, ορίζεται στο αρχείο zabbix_agentd.conf για κάθε παράγοντα. Αυτό φαίνεται στο παρακάτω διάγραμμα:



Εκτός από την έναρξη της σύνδεσης στο δίκτυο, η κύρια διαφορά μεταξύ παθητικού στοιχείου και ενός ενεργού στοιχείου είναι ότι, στο τελευταίο, είναι αδύνατο να οριστούν ευέλικτα διαστήματα παρακολούθησης. Με ένα παθητικό στοιχείο, μπορεί να οριστεί διαφορετική παρακολούθηση σε διαστήματα ανάλογα με την ώρα της ημέρας και την ημέρα της εβδομάδας. Για παράδειγμα, εσείς θα μπορούσατε να ελέγξετε τη διαθεσιμότητα ενός server διαχείρισης ταυτότητας κάθε λεπτό κατά τη διάρκεια ωρών γραφείου και κάθε 10 λεπτά κατά τη διάρκεια της νύχτας. Από την άλλη, εάν χρησιμοποιείτε ένα ενεργό στοιχείο, έχετε κολλήσει με μία μόνο επιλογή για την παρακολούθηση των διαστημάτων. Μπορεί επίσης να έχετε παρατηρήσει μια ομοιότητα μεταξύ του Zabbix ενεργών και παθητικών στοιχείων και την ομοιότητα μεταξύ των λειτουργιών των ενεργών και παθητικών Zabbix.

Στην πραγματικότητα, μπορείτε να επιλέξετε μεταξύ των ενεργών και των παθητικών αντικειμένων με τον ίδιο τρόπο και για τους ίδιους λόγους που επιλέγετε μεταξύ ενεργού ή παθητικού διακομιστή μεσολάβησης για να αφαιρέσετε ορισμένες από τις εργασίες

προγραμματισμού του server και για να επιλύσετε τους περιορισμούς και τους περιορισμούς του δικτύου σας και τη δρομολόγηση ή διαμόρφωση τείχους προστασίας.

Υπάρχει, βεβαίως, μια κύρια διαφορά μεταξύ proxies και agents. Ένας proxy δεν μπορεί να συλλέξει δεδομένα παρακολούθησης από πολλούς διαφορετικούς hosts, ενώ ένας agent είναι θεωρητικά για την παρακολούθηση μόνο του host που είναι εγκατεστημένος.

Η κύρια διαφορά όταν πρόκειται για τη ροή δεδομένων, είναι ότι ο τρόπος λειτουργίας του proxy εφαρμόζεται σε όλους τους hosts και τα items που διαχειρίζεται ο proxy. Στην πραγματικότητα, αυτός δεν ενδιαφέρεται για τη φύση των αντικειμένων που πρέπει να παρακολουθεί ένας proxy. Ωστόσο, όταν ένας ενεργός proxy συγκεντρώνει τα δεδομένα του (είτε με στοιχεία ενεργών είτε παθητικών στοιχείων, εξωτερικά scripts, SNMP, SSH κ.ο.κ.), θα ξεκινήσει πάντα όλες τις συνδέσεις με το server.

Το ίδιο ισχύει και για έναν παθητικό proxy. Δεν έχει σημασία αν όλα τα itemsπου πρέπει να παρακολουθούνται είναι ενεργά items του agent. Πάντα θα περιμένει τον server για ενημερώσεις ρυθμίσεων και μετρήσεων.

Από την άλλη πλευρά, ένα ενεργό ή παθητικό item είναι απλώς ένα item όπως όλα τα υπόλοιπα. Ένας host μπορεί να ορίζει συνδυασμούς μεταξύ ενεργών και παθητικών στοιχείων. Επομένως, δεν μπορείτε να υποθέσετε ότι ένας agent ξεκινάει όλες τις συνδέσεις του με τον server. Για να γίνει αυτό, όλα τα στοιχεία που βασίζονται στον agent πρέπει να οριστούν ως ενεργά, συμπεριλαμβανομένων των μελλοντικών.

Zabbixtrapperitems

Μια ακραία έκδοση ενός ενεργού στοιχείου που εξακολουθεί να βασίζεται στο πρωτόκολλο του zabbixagent είναι το trapper στοιχείο του Zabbix. Είναι ο μοναδικός μεταξύ όλων των άλλων τύπων items σε ενατrapperitem όπου δεν υπάρχει δυνατότητα ορισμού χρονικού διαστήματος παρακολούθησης στο επίπεδο του server. Με άλλα λόγια, ένας server θα γνωρίζει αν έχει οριστεί κάποιο trapperitem, ο τύπος δεδομένων του, ο host που σχετίζεται με αυτόν, και την περίοδο διατήρησης τόσο για το ιστορικό όσο και για τις τάσεις. Αλλά δεν θα προγραμματίσει ποτέ τον έλεγχο για το στοιχείο ούτε θα περάσει η πληροφορία προγραμματισμού και διαστήματος παρακολούθησης σε οποιονδήποτε proxy ή agent. Έτσι, εξαρτάται από τον

συγκεκριμένο ανιχνευτή να προγραμματιστεί με κάποιο τρόπο και στη συνέχεια, να στείλει τις πληροφορίες σχετικά με τα δεδομένα που έχει συγκεντρώσει, στον server.

Τα trapper στοιχεία είναι το αντίθετο από τους εξωτερικούς ελέγχους του Zabbix από άποψη ροής δεδομένων. Ένα στοιχείο τύπου εξωτερικού ελέγχου ορίζεται όταν θέλουμε ο server να εκτελέσει ένα εξωτερικό script για να συγκεντρώσει μετρήσεις αντί να τις ζητήσει από έναν agent (Zabbix, SNMP ή άλλους). Αυτό μπορεί να φορτώσει αρκετά την απόδοση του server καθώς πρέπει να εκτελεί μια νέα διαδικασία για κάθε εξωτερικό script που πρέπει να εκτελέσει και στη συνέχεια να περιμένει την απάντηση. Όσο ο αριθμός των εξωτερικών script αυξάνεται, μπορεί να επιβραδύνει σημαντικά τις λειτουργίες του server μέχρι να συσσωρεύσει έναν μεγάλο αριθμό καθυστερημένων ελέγχων, ενώ είναι απασχολημένος να εκτελεί scripts. Ένας εξαιρετικά απλός και πρωτόγονος, αλλά αποτελεσματικός, τρόπος αντιμετώπισης αυτού του προβλήματος (μετά

μείωση του αριθμού των εξωτερικών σεναρίων, όσο το δυνατόν περισσότερο φυσικά) είναι η μετατροπή όλων των στοιχείων εξωτερικού ελέγχου σε trapper στοιχεία, προγραμματίζοντας την εκτέλεση των scripts, χρησιμοποιώντας crontab ή κάποιο άλλο εργαλείο προγραμματισμού και μετατρέποντας τα έτσι ώστε να χρησιμοποιούν την εντολή zabbix_sender και να στέλνει τις μετρήσεις τον zabbixserver.

Επισκόπηση ροής δεδομένων

Πρόκειται για μια ανασκόπηση των τύπων στοιχείων του zabbix που ταξινομούνται με τον τύπο σύνδεσης με μια προτεινόμενη εναλλακτική λύση που πιθανώς κάποιος να θέλει να αλλάξει. Όπως μπορείτε να δείτε, τα trapper στοιχεία είναι συχνά η μόνη πιθανή εναλλακτική λύση αν κάποιος έχει την ανάγκη να αντιστρέψει τον τύπος σύνδεσης. Σημειώστε ότι στον ακόλουθο πίνακα, ο όρος Passive σημαίνει ότι η σύνδεση ξεκινάει από το server και Active ότι η σύνδεση ξεκινάει από οποιονδήποτεclient.

Item Type	Direction	Alternative
Zabbix agent	Passive	Zabbix agent(active)
Zabbix agent (active)	Active	Zabbix agent
Simple check	Passive	Zabbix trapper
SNMP agent	Passive	Zabbix trapper (SNMP traps are completely different in nature)
SNMP trap	Active	N/A
Zabbix internal	N/A (data about the server monitoring itself)	N/A
Zabbix trapper	Active	Depends on the nature of the monitored data
Zabbix aggregate	N/A (uses data already available in the database)	N/A
External check	Passive	Zabbix trapper
Database monitor	Passive	Zabbix trapper
IPMI agent	Passive	Zabbix trapper
SSH agent	Passive	Zabbix trapper
TELNET agent	Passive	Zabbix trapper
JMX agent	Passive	Zabbix trapper
Calculated	N/A (uses data already in the database)	N/A

ValueMapping

Για μια πιο «ανθρώπινη» αναπαράσταση των τιμών που λαμβάνετε, μπορείτε να χρησιμοποιήσετε τα valuemaps που περιέχουν αντιστοίχιση μεταξύ αριθμητικών τιμών και συμβολοσειρών. Οι αντιστοιχίσεις τιμών μπορούν να χρησιμοποιηθούν τόσο στο γραφικό περιβάλλον του Zabbix όσο και στις ειδοποιήσεις που αποστέλλονται με email / SMS / jabber κ.λπ.

Για παράδειγμα, ένα στοιχείο που έχει τιμή «0» ή «1» μπορεί να χρησιμοποιήσει το valuemapping για να αναπαραστήσει τις τιμές σε μορφή αναγνώσιμη από τον άνθρωπο:

'0' ⇒'Μη διαθέσιμο'

'1' ⇒'Διαθέσιμο'

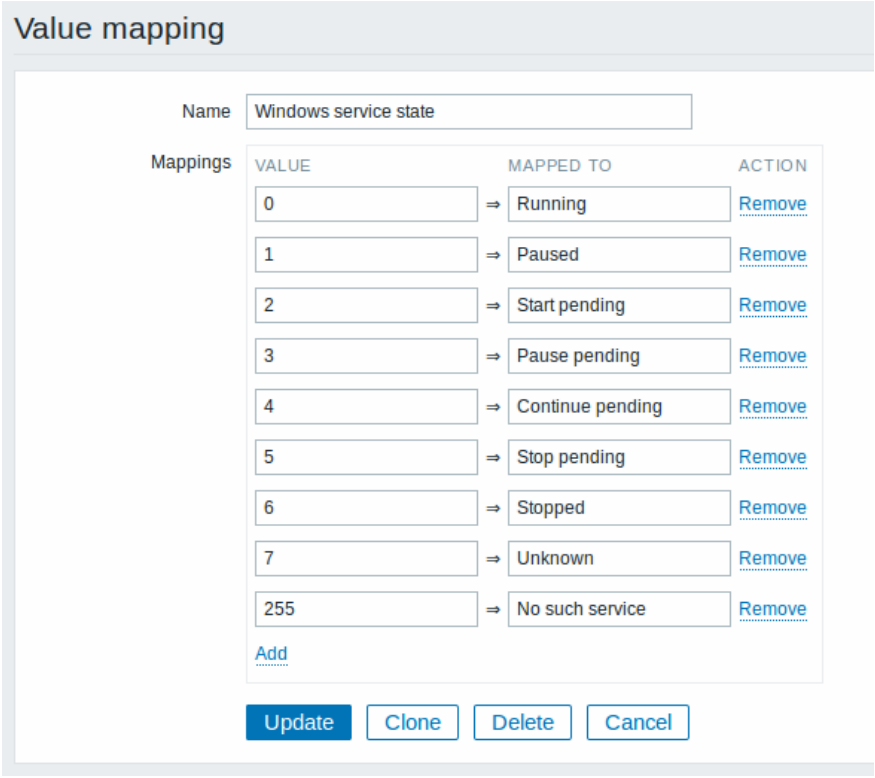
Ή, ένα σχετικό valuemapping μπορεί να είναι:

'F' → 'Πλήρης'

'D' → 'Διαφορικό'

'I' → 'Incremental'

Ένας άλλος λόγος που χρησιμοποιούμε το valuemapping είναι για να εξοικονομίσουμε χώρο στη βάση δεδομένων. Σκεφτείτε να διαχειρίζεστε μια τεράστια υποδομή και οι τιμές που επιστρέφουν τα χιλιάδες items να ήταν συμβολοσειρές. Στην παρακάτω εικόνα απεικονίζεται ο πίνακας του valuemapping



UserParameters

Κάποιες φορές, ίσως θέλετε να τρέξετε κάποιον έλεγχο ο οποίος δεν υπάρχει σε κάποιο item του Zabbix. Είναι και ο λόγος που υπάρχουν τα UserParameters. Με λίγα λόγια, μπορείτε να γράψετε κάποια εντολή η οποία επιστρέφει τα δεδομένα που χρειάζεστε, και να τη συμπεριλάβετε στο αρχείο ρυθμίσεων του agent που θέλετε.

Τα userparameters την ακόλουθη σύνταξη:

UserParameter=<key>,<command>

Όπως μπορείτε να δείτε, ένα UserParameter περιέχει επίσης ένα key. Το key είναι απαραίτητο κατά τη διαμόρφωση ενός στοιχείου. Εισάγετε ένα key της επιλογής σας, το οποίο θα είναι εύκολο να αναφερθεί (πρέπει να είναι μοναδικό σε host). Στη συνέχεια θα χρειαστεί να επανεκκινήσετε τον zabbixagent για να πάρει τις αλλαγές που έχουν γίνει στο αρχείο ρυθμίσεων. Στη συνέχεια, κατά τη διαμόρφωση ενός item, πληκτρολογήστε το κλειδί για να αναφερθείτε στην εντολή από το UserParameter που θέλετε να εκτελεστεί. Τα UserParameters είναι εντολές που εκτελούνται από τον Zabbixagent. Μπορούν να επιστραφούν έως και 512KB δεδομένα. Ωστόσο, ό,τι τιμή κειμένου μπορεί τελικά να αποθηκευτεί στη βάση δεδομένων περιορίζεται σε 64KB στην MySQL.
Το **/bin/sh** χρησιμοποιείται ως διερμηνέας γραμμής εντολών στα λειτουργικά συστήματα UNIX. Οι παράμετροι χρηστών ακολουθούν το χρονικό όριο ελέγχου του παράγοντα. Εάν λήξει η χρονική υπέρβαση, η διαδικασία παραμέτρου χρήστη διακόπτεται.

Ακολουθεί ένα παράδειγμα μιας απλής εντολής:
UserParameter=ping,echo 1
Πάντα θα επιστρέφει πάντα 1 για κάθε item με key ‘ping’

Ένα πιο περίπλοκο παράδειγμα:
UserParameter=mysql.ping,mysqladmin -urootping | grep -c alive
Ο agent θα επιστρέφει πάντα , αν ο MySQLserver λειτουργεί. Αλλιώς επιστρέφει 0.

Απεικόνιση Δεδομένων Zabbix

Μόλις το zabbix εγκατασταθεί σε μια υποδομή, είναι έτοιμο να υποστηρίξει ένα μεγάλο φόρτο εργασίας και να διαχειριστεί έναν τεράστιο όγκο δεδομένων. Το επόμενο βήμα είναι η γραφική παράσταση των δεδομένων, η παρεμβολή και η συσχέτιση των μετρήσεων μεταξύ τους. Το ισχυρό σημείο είναι ότι μπορείτε να συσχετίσετε διαφορετικό τύπο μετρήσεων στον ίδιο άξονα

του χρόνου, να αναλύσετε μοντέλα βαριάς και ελαφριάς χρήσης, προσδιορίζοντας τις υπηρεσίες και τον εξοπλισμό που αποτυγχάνουν πιο συχνά στην υποδομή σας, και να αποτυπώσετε τις σχέσεις μεταξύ των μετρήσεων των συνδεδεμένων υπηρεσιών.
Πέρα από την εγκατάσταση των τυπικών γραφημάτων, το Zabbix προσφέρει έναν τρόπο να δημιουργήσετε το δικό σας γράφημα και να το προσθέσετε στο δικό σας πρότυπο, δημιουργώντας έτσι μια εύκολη μέθοδο να διαδώσετε τα γραφήματα σας σε όλους τους servers. Αυτά τα προσαρμοσμένα γραφήματα μπορούν να συλλεχθούν σε screens. Μέσα στο Zabbix, ένα screen μπορεί να περιέχει διαφορετικά είδη πληροφοριών - απλά γραφήματα, προσαρμοσμένα γραφήματα, άλλα screen, πληροφορίες απλού κειμένου, επισκοπήσεις των triggers και ούτω καθεξής.

Ως πρακτικό παράδειγμα, μπορείτε να σκεφτείτε ένα μεγάλο κέντρο δεδομένων, όπου υπάρχουν διαφορετικά επίπεδα υποστήριξης. Συνήθως, το πρώτο επίπεδο υποστήριξης πρέπει να έχει μια γενική επισκόπηση του τι συμβαίνει στο κέντρο δεδομένων, το δεύτερο επίπεδο είναι χωρισμένο σε τμήματα πχ DBA, servers λογισμικού και ούτω καθεξής. Το τμήμα DBA θα χρειαστεί πληροφορίες από τις μετρήσεις των βάσεων δεδομένων, ενώ ένας ειδικός στους servers λογισμικού πιθανότατα θα χρειάζεται όλες τις μετρήσεις Java, καθώς και κάποιες άλλες τυπικές μετρήσεις, όπως η μνήμη και χρήση CPU. Οι απαντήσεις του Zabbix σε αυτήν την απαίτηση είναι τα maps, τα screens και τα slides.
Μόλις δημιουργηθούν όλα τα γραφήματα και έχουν ανακτηθεί όλες μετρήσεις και τα μηνύματα που χρειάζεται κάποιος, μπορεί κάποιος να δημιουργήσει εύκολα screens που συλλέγουν, για παράδειγμα, όλα τα γραφήματα σχετικά με τις βάσεις δεδομένων συν μερικές άλλες τυπικές μετρήσεις. Τascreens μπορούν να συλλεχθούν σε slides και να καλύψουν κάθε επίπεδο υποστήριξης, βλέποντας τα screens ομαδοποιημένα σε μια παρουσίαση, η οποία έχει άμεση ποιοτική και ποσοτική επισκόπηση για το τι συμβαίνει.

Τα γραφήματα είναι ισχυρά εργαλεία για τη μετάδοση του μηνύματός σας. Είναι ένα ευέλικτο όργανο που μπορεί να χρησιμοποιηθεί για να δώσει περισσότερη δύναμη στην ομιλία σας, καθώς και να δώσει μια ποιοτική επισκόπηση της υπηρεσίας ή της υποδομής σας.

Απεικόνιση μέσω screens

Μια οθόνη είναι μια σελίδα που μπορεί να εμφανίσει πολλαπλά στοιχεία zabbix, όπως γραφήματα, χάρτες και κείμενο. Μία από τις κύριες διαφορές μεταξύ των screens και των maps είναι ότι σε maps, μπορείτε να τοποθετήσετε πολλά στοιχεία, αλλά εσείς δεν μπορείτε, για παράδειγμα, να προσθέσετε ένα γράφημα ή την κατάσταση του trigger. Έχουν δύο διαφορετικούς στόχους. Η οθόνη μπορεί να ομαδοποιήσει όλα τα στοιχεία που είναι κοινά σε ένα συγκεκριμένο είδος server για να έχετε μια πλήρη εικόνα της κατάστασης.

Δείτε ένα screen στην παρακάτω εικόνα:

Απεικόνιση των δεδομένων μέσω maps

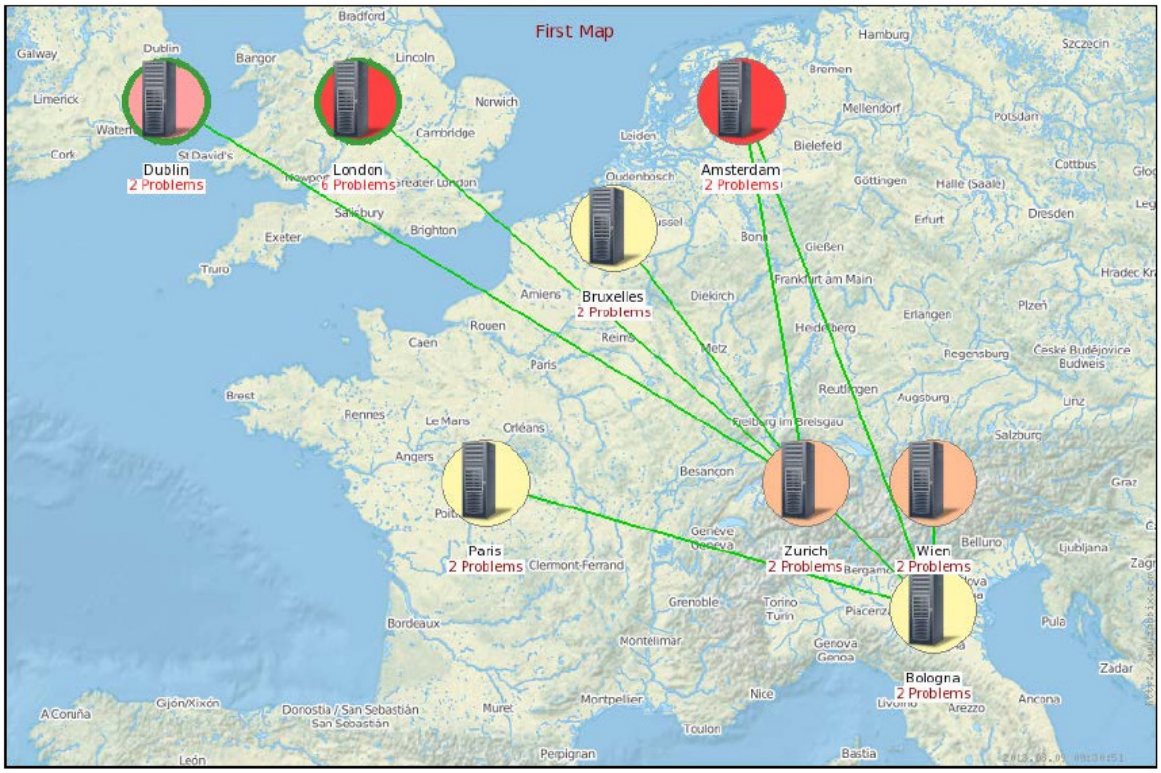
Το Zabbix παρέχει ένα ισχυρό στοιχείο για την απεικόνιση δεδομένων και μια τοπολογική προβολή Zabbix, το οποίο βοηθάει στη δημιουργία χαρτών. Οι χάρτες είναι μια γραφική

παράσταση της φυσικής σας υποδομής, όπου μπορείτε να εμφανίσετε τους servers σας, τον δικτυακό εξοπλισμό και τη διασύνδεση μεταξύ τους.
Το μεγάλο πλεονέκτημα είναι ότι οι χάρτες στο Zabbix είναι εντελώς δυναμικοί, πράγμα που σημαίνει ότι θα δείτε τις ενεργές προειδοποιήσεις, τα θέματα και triggers που αναπαρίστανται στο χάρτη, με διαφορετικά εικονίδια, χρώματα και ετικέτες. Είναι μια ισχυρή αναπαράσταση των datacenters ή των υπηρεσίων.

Τα στοιχεία που μπορείτε να βάλετε σε ένα map είναι οι εξής:

- Host
- Hostgroups
- Triggers
- Images
- Maps

Στην παρακάτω εικόνα απεικονίζεται Zabbixmap:



Triggers

Τα triggers, εξετάζουν τα δεδομένα των items και ειδοποιούν το σύστημα όταν τα δεδομένα δεν ταιριάζουν με όποια προϋπόθεση έχει οριστεί. Η συλλογή δεδομένων είναι καλή, αλλά εξαιρετικά ανεπαρκής. Αν θέλετε προηγούμενες συλλογές ιστορικών δεδομένων, συμπεριλαμβανομένων των ειδοποιήσεων, θα έπρεπε να υπάρχει κάποιος που θα εξετάζει όλα τα δεδομένα όλη την ώρα, οπότε πρέπει να ορίσουμε κατώτατα όρια, με τα οποία θέλουμε να εξεταστεί η κατάσταση. Τα triggers παρέχουν έναν τρόπο για να καθορίσουν ποιες είναι αυτές οι συνθήκες.

Όπως βλέπετε στην παρακάτω εικόνα, για τη δημιουργία ενός trigger πρέπει να συμπληρωθούν τα εξής πεδία:

- Name: Τοπεριεχόμενο αυτού του πεδίου θα χρησιμοποιείται για τον εντοπισμό του trigger στα περισσότερα μέρη, οπότε θα πρέπει να είναι ευανάγνωστο και να δηλώνει την κατάσταση που θα προκύψει(πχ. SMPT serviceisDown).

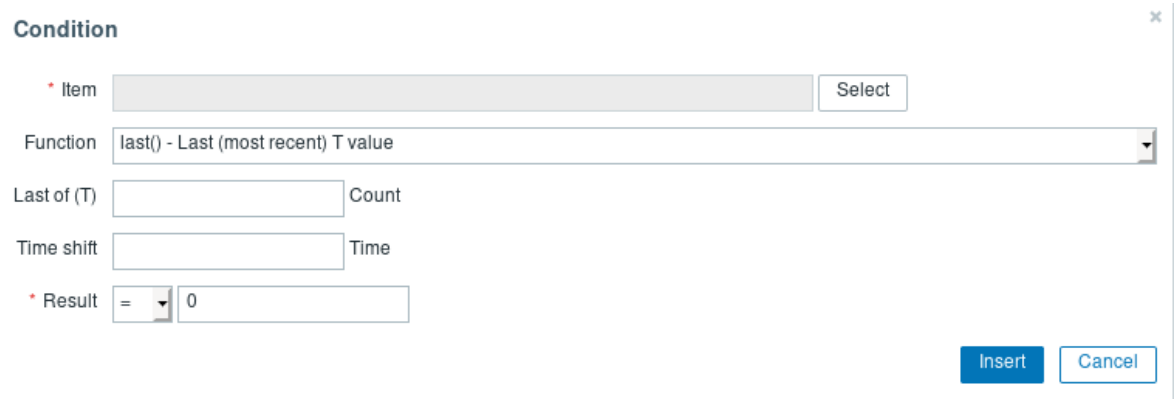
- Expression: Αυτή είναι η πιο σημαντική ιδιότητα ενός trigger. Εδώ καθορίζονται οι συνθήκες ελέγχων.

  Item: Εδώ μπορούμε να καθορίσουμε ποια δεδομένα στοιχείων θα πρέπει να ελεγχθούν.

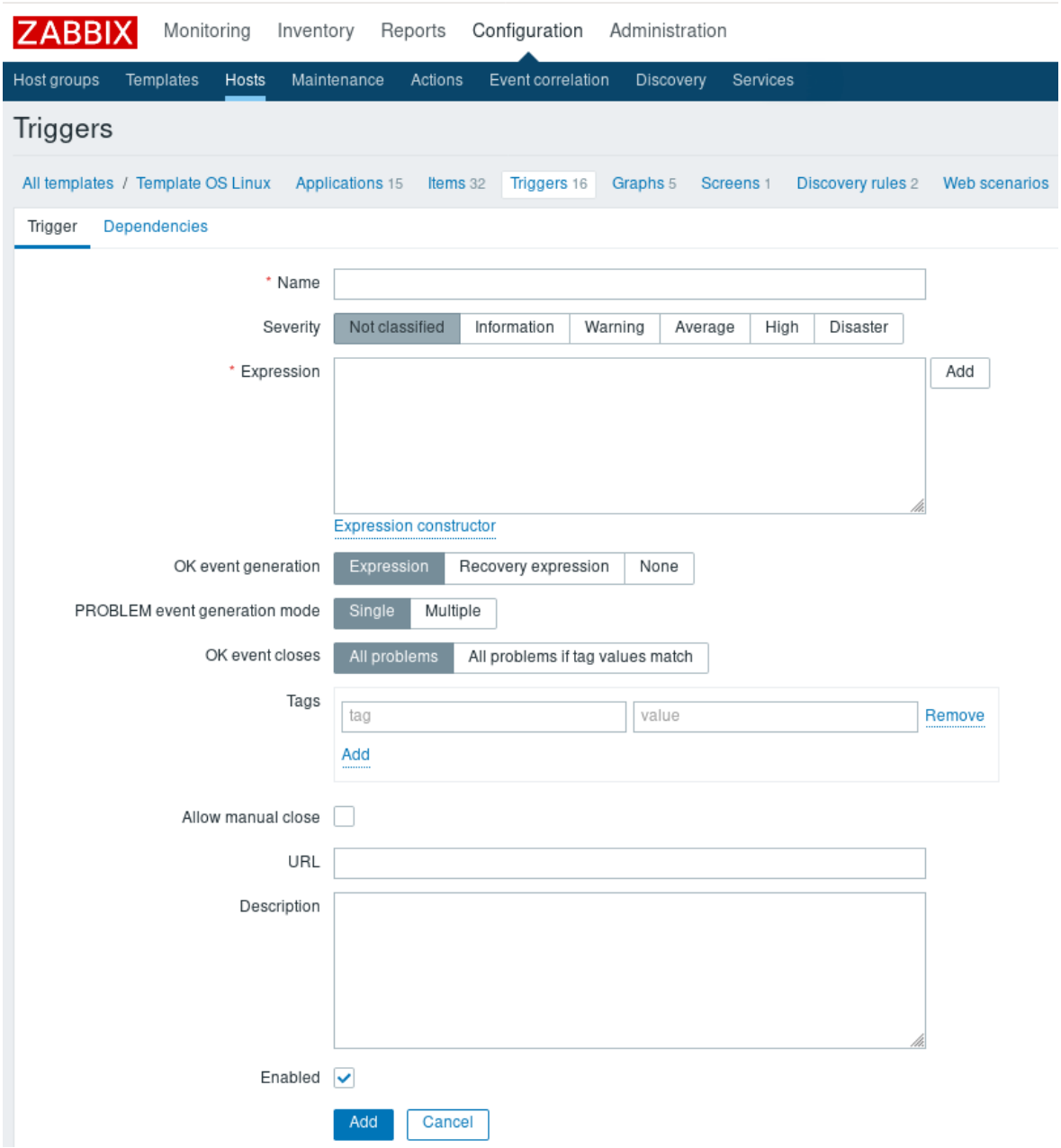
  Function: Εδώ μπορούμε να δηλώσουμε την συνθήκη για την οποία θα ενεργοποιηθεί το trigger.

  Listof (T): Δηλώνουμε την παράμετρο αυτή εάν η λειτουργία υποστηρίζει μια χρονική περίοδο.

  N: Αυτό το πεδίο μας επιτρέπει να ορίσουμε μια σταθερά



- Severity: Υπάρχουν πέντε επίπεδα σοβαρότητας στο Zabbix και μια επιπλέον μη κατηγοριοποιημένη, όπως φαίνεται εδώ:



Ο έλεγχος των συνθηκών και των συναγερμών είναι η πιο χαρακτηριστική λειτουργία κάθε συστήματος παρακολούθησης και το Zabbix δεν αποτελεί εξαίρεση. Αυτό που κάνει το Zabbix ξεχωριστό είναι ότι κάθε κατάσταση ή trigger (όπως είναι γνωστό σε αυτό το σύστημα) μπορεί να συνδεθεί όχι μόνο με μία μόνο μέτρηση, αλλά και με έναν αυθαίρετο πολύπλοκο υπολογισμό βασισμένο σε όλα τα δεδομένα που είναι διαθέσιμα στο διακομιστήZabbix. Επιπλέον, έτσι όπως τα triggers είναι ανεξάρτητα από στοιχεία, οι ενέργειες που μπορεί να λάβει ο διακομιστής με βάση την κατάσταση ενεργοποίησης είναι ανεξάρτητες από το μεμονωμένο trigger.

Μια αποτελεσματική, σωστή και ολοκληρωμένη διαμόρφωση προειδοποίησης αποτελεί το κλειδί της επιτυχίας ενός συστήματος παρακολούθησης. Βασίζεται σε εκτεταμένη συλλογή δεδομένων και τελικά οδηγεί στη διαχείριση μηνυμάτων, παραληπτών, και μέσα αποστολής, όπως θα δούμε αργότερα στο κεφάλαιο.

Σοβαρότητα trigger

Η σοβαρότητα είναι κάτι παραπάνω από μια απλή ετικέτα που επισυνάπτεται σε ένα trigger. Το γραφικό περιβάλλον εμφανίζει διαφορετικές τιμές σοβαρότητας με διαφορετικά χρώματα και το κάνει να είναι σε θέση να δημιουργήσει διαφορετικές δράσεις βασισμένες σε αυτές, αλλά δεν έχουν περαιτέρω νόημα ή λειτουργία στο σύστημα. Αυτό σημαίνει ότι η σοβαρότητα μιας πυροδότησης δεν θα αλλάξει με την πάροδο του χρόνου, βασισμένη στο πόσο καιρό η πυροδότηση αυτή βρίσκεται σε μια προβληματική κατάσταση, ούτε μπορείτε ορίσετε μια διαφορετική σοβαρότητα σε διαφορετικά όρια στην ίδια πυροδότηση. Αν πραγματικά χρειάζεστε μια προειδοποίηση όταν ένας δίσκος είναι πάνω από 90% πλήρης και μια κρίσιμη ειδοποίηση όταν είναι 100% πλήρης, θα χρειαστεί να δημιουργήσετε δύο διαφορετικές πυροδοτήσεις με δύο διαφορετικά κατώτατα όρια και σοβαρότητες. Αυτό μπορεί να μην είναι η καλύτερη πορεία ενεργειών, καθώς θα μπορούσε να οδηγήσει σε προειδοποιήσεις που αγνοούνται και δεν έλαβαν χώρα, να ενεργοποιηθούν κρίσιμες προειδοποιήσεις όταν είναι ήδη πολύ αργά και να χάσετε τη διαθεσιμότητα υπηρεσιών. Μια καλύτερη προσέγγιση θα ήταν να εκτιμήσετε με σαφήνεια και σοβαρότητα τη πιθανότητα να γεμίσει ο δίσκος και να δημιουργήσετε μια μόνο πυροδότηση με μια λογική σταθερά και, ενδεχομένως, μια κλιμακωτή ενέργεια εάν φοβάστε ότι η προειδοποίηση θα μπορούσε να χαθεί μεταξύ των άλλων.

Λήψη ενεργειών

Ακριβώς όπως τα στοιχεία παρέχουν μόνο ανεπεξέργαστα δεδομένα και τα triggers είναι ανεξάρτητα από αυτά, μπορούν να αποκτήσουν πρόσβαση σε ιστορικά δεδομένα σχεδόν κάθε item, τα triggers, με τη σειρά τους, παρέχουν μόνο μια κατάσταση αλλαγής. Αυτή η αλλαγή καταγράφεται ως συμβάν ακριβώς όπως οι μετρήσεις καταγράφονται ως δεδομένα items. Αυτό

σημαίνει ότι τα triggers δεν παρέχουν λειτουργίες αναφοράς. Απλά ελέγχουν τις συνθήκες τους και αλλάζουν ανάλογα την κατάσταση. Και πάλι, μπορεί να φαίνεται να είναι περιορισμός και η έλλειψη δύναμης αποδεικνύεται ότι είναι το ακριβώς αντίθετο με το Zabbixitem είναι υπεύθυνο για την πραγματική αποστολή ειδοποιήσεων ή για την αυτόματη προσπάθεια η επίλυση ορισμένων προβλημάτων είναι απολύτως ανεξάρτητη από τους μηχανισμούς ενεργοποίησης. Αυτό σημαίνει οι ενεργοποιητές μπορούν να έχουν πρόσβαση σε δεδομένα οποιουδήποτε στοιχείου, οι ενέργειες μπορούν να έχουν πρόσβαση σε οποιοδήποτε όνομα ενεργοποίησης, σοβαρότητα, ή την κατάσταση, έτσι ώστε, για άλλη μια φορά, να δημιουργήσετε το τέλειο μείγμα πολύ γενικών και πολύ συγκεκριμένες ενέργειες χωρίς να είναι κολλημένη σε ένα σύστημα δράσης ανά σκανδαλισμό.

Σε αντίθεση με τις πυροδοτήσεις, οι ενέργειες είναι επίσης εντελώς ανεξάρτητες από τους hosts και τα πρότυπα. Κάθε ενέργεια ορίζεται πάντοτε σε παγκόσμια κλίμακα και οι συνθήκες της ελέγχονται με κάθε συμβάν του Zabbix.

Μια ενέργεια αποτελείται από τρία διαφορετικά μέρη που συνεργάζονται μαζί και παρέχουν όλες τις απαραίτητες λειτουργίες:

- Ορισμός δράσης
- Συνθήκες δράσης
- Ενέργειες δράσης

Πρότυπα

Για όλη την ισχύ παρακολούθησης των αντικειμένων, γραφημάτων, χαρτών και πυροδοτήσεων του Zabbix, θα ήταν απίστευτα άβολο να δημιουργήσετε με μη αυτόματο τρόπο κάθε ένα από αυτά τα αντικείμενα για κάθε παρακολουθούμενο κεντρικό υπολογιστή. Στην περίπτωση ενός μεγάλου περιβάλλοντος, με εκατοντάδες ή χιλιάδες αντικείμενα που παρακολουθούνται, θα ήταν πρακτικά αδύνατο να γίνει χειροκίνητα να ρυθμίσετε όλα τα στοιχεία, τα γραφήματα και τις πυροδοτήσεις που απαιτούνται για κάθε κεντρικό υπολογιστή.

Χρησιμοποιώντας τη λειτουργία προτύπων, ορίζετε διαφορετικές συλλογές αντικειμένων, πυροδοτήσεων και γραφημάτων προκειμένου να εφαρμοστούν κοινές ρυθμίσεις σε οποιονδήποτε αριθμό κεντρικών υπολογιστών, ενώ ακόμα είναι σε θέση να διαχειριστεί κάθε πτυχή που μπορεί να χρειαστείτε να κάνετε κάποια μικροδιόρθωση για κάθε έναν κεντρικό υπολογιστή.

Το τέλειο συμπλήρωμα της εγκατάστασης του προτύπου είναι το χαρακτηριστικό της ανακάλυψης του Zabbix. Χρησιμοποιώντας αυτό, θα καθορίσετε ένα σύνολο κανόνων που θα επιτρέπουν στον Zabbix να γνωρίζει νέους hosts χωρίς να χρειάζεται τα δημιουργήσετε με το χέρι. Μπορείτε επίσης να επωφεληθείτε από την ανακάλυψη χαμηλού επιπέδου(lowleveldescovery) ώστε να μπορείτε να αντιστοιχίσετε αυτόματα τα σωστά στοιχεία ακόμη και για εκείνα τα ιδιαίτερα μεταβλητά μέρη ενός συστήματος, όπως ο αριθμός και ο τύπος των δίσκων, συστήματα αρχείων και προσαρμογείς δικτύου.

Δημιουργία προτύπων

Το πρότυπο ενός διακομιστή έχει παρόμοια λειτουργία με εκείνη ένος συνηθισμένου zabbixhost. Και τα δύο είναι συλλογές αντικειμένων, πυροδοτήσεων, γραφήματων, οθονών και κανόνων ανακάλυψης χαμηλού επιπέδου. Και τα δύο χρειάζονται ένα μοναδικό όνομα όπως και κάθε άλλη οντότητα στο Zabbix. Και τα δύο μπορούν να ανήκουν σε μία ή περισσότερες ομάδες. Η κρίσιμη διαφορά είναι ότι ένας zabbixagent έχει ένα ή περισσότερα πρότυπα για να έλθει σε επαφή, έτσι ώστε ο διακομιστήςZabbix να μπορεί στην πραγματικότητα να λάβει μετρήσεις στοιχείων απο αυτό. Αυτές μπορεί να είναι μία ή περισσότερες διευθύνσεις IP ή DNS που αντιπροσωπεύουν zabbixinterfaces ή SNMP, JMX και IPMI. Έτσι, ένας zabbixhost είναι ένα αντικείμενο του διακομιστήZabbix θα ζητήσει πληροφορίες ή θα περιμένει να λάβει δεδομένα. Ένα πρότυπο, από την άλλη, δεν διαθέτει διεπαφή πρόσβασης, οπότε ο διακομιστήςZabbix δεν θα προσπαθήσει ποτέ να ελέγξει αν ένα πρότυπο είναι ζωντανό ή να ζητήσει τελευταίες μετρήσεις αντικειμένων.

ZabbixLowLevelDiscovery

Ένα ακόμη πιο χρήσιμο και σημαντικό χαρακτηριστικό των προτύπων του Zabbix είναι η ικανότητά του να υποστηρίξει ειδικά είδη αντικειμένων, τα οποία ονομάζονται κανόνες χαμηλού επιπέδου ανακάλυψης. Μόλις εφαρμοστούν σε πραγματικούς zabbixhosts, αυτοί οι κανόνες θα

ζητήσουν απο τον zabbixhost οποιοδήποτε είδος απο πόρους που έχουν ρυθμιστεί να αναζητούν: συστήματα αρχείων, διεπαφές δικτύου, SNMP OID και πολλά άλλα. Για κάθε πόρο που βρέθηκε, ο διακομιστής θα δημιουργήσει δυναμικά στοιχεία, πυροδοτήσεις, και γραφήματα σύμφωνα με ειδικά πρωτότυπα οντοτήτων που συνδέονται με το κανόνες ανακάλυψης.

Το μεγάλο πλεονέκτημα των κανόνων ανακάλυψης χαμηλού επιπέδου είναι ότι φροντίζουν τα περισσότερα μεταβλητά μέρη ενός παρακολουθούμενου κεντρικού υπολογιστή, όπως ο τύπος και ο αριθμός των προσαρμογέων δικτύου, με έναν δυναμικό και γενικό τρόπο. Αυτό σημαίνει ότι, αντί να δημιουργείτε με το χέρι συγκεκριμένα στοιχεία και πυροδοτήσεις για κάθε προσαρμογέα δικτύου ή συστήματος αρχείων ενός zabbixhost ή να δημιουργήσετε τεράστια πρότυπα με κάθε πιθανό είδος αντικειμένου για μια συγκεκριμένη λειτουργία συστήματος και διατηρώντας τα περισσότερα από αυτά απενεργοποιημένα, μπορείτε να έχετε έναν λογικό αριθμό γενικών προτύπων που θα προσαρμοστούν στις ιδιαιτερότητες κάποιου συγκεκριμένου zabbixhost δημιουργώντας άμεσα κάθε οντότητα που απαιτείται βάσει των ανακαλυφθέντων πόρων και των προηγουμένως διαμορφωμένων πρωτύπων.

Από μόνο του, το Zabbix υποστηρίζει τέσσερις κανόνες ανακάλυψης:

- Διεπαφές δικτύου
- Τύποι συστημάτων αρχείων
- SNMP OIDs
- CPU και πυρήνες CPU

Καθώς οι κανόνες ανακάλυψης είναι πραγματικά ένα ειδικό είδος αντικειμένου, μπορείτε να δημιουργήσετε τους δικούς σας κανόνες, υπό τον όρο ότι καταλαβαίνετε την ιδιαιτερότητά τους σε σχέση με τα κανονικά αντικείμενα. Ένα στοιχείο ανακάλυψης επιστρέφει πάντα μια λίστα ζευγών μακροεντολών / τιμών που εκφράζονται σε JSON. Αυτή η λίστα αναπαριστά όλους τους πόρους που βρέθηκαν από τα στοιχεία ανακάλυψης μαζί με τη σημασία της αναφοράς τους. Ο παρακάτω πίνακας εμφανίζει τα αντικείμενα ανακάλυψης του Zabbix και τις τιμές που επιστρέφουν μαζί με μια γενίκευση που θα σας δώσει μια ιδέα για το πώς να δημιουργήσετε τους

δικούς σας κανόνες:

Discovery item key	Item type	Return values
net.if.discovery	Zabbix agent	{"data":[{"{#IFNAME}":"<name>"}, {"{#IFNAME}":"<name>"}, {"{#IFNAME}":"<name>"}, …]}
system.cpu.discovery	Zabbix agent	{"data": [{"{#CPU.NUMBER}":"<idx>", "{#CPU.STATUS}":"<value>"}, {"{#CPU.NUMBER}":"<idx>", "{#CPU.STATUS}":"<value>"}, {"{#CPU.NUMBER}":"<idx>", "{#CPU.STATUS}":"<value>"}, …] }
snmp.discovery	SNMP (v1, v2, or v3) agent	{"data":[{"{#SNMPINDEX}":"<idx>", "{#SNMPVALUE}":"<value>}, {"{#SNMPINDEX}":"<idx>", "{#SNMPVALUE}":"<value>}, {"{#SNMPINDEX}":"<idx>", "{#SNMPVALUE}":"<value>}, …]}
custom.discovery	Any	{"data":[{"{#CUSTOM1}":"<value>","{#CUSTOM2}":"<value>"}, {"{#CUSTOM1}":"<value>","{#CUSTOM2}":"<value>"}, {"{#CUSTOM1}":"<value>","{#CUSTOM2}":"<value>"}, …]}

Χειρισμός εξωτερικών σεναρίων

Μέχρι τώρα, είδατε πώς λειτουργούν τα περισσότερα από τα στοιχεία ενός διακομιστή και πώς αναγκάστε το Zabbix να αποκτήσει δεδομένα από διάφορες εξωτερικές πηγές.

Πιθανότατα, θα βρείτε μια διαφορετική προσαρμοσμένη συσκευή, συσκευή διακομιστή και ιδιόκτητο υλικό. Συνήθως, όλες αυτές οι συσκευές έχουν μια διεπαφή που πρέπει να διερευνηθεί, αλλά, δυστυχώς, συμβαίνει συχνά όταν οι περισσότερες μετρήσεις δεν εκτίθενται

μέσω του Πρωτοκόλλου διαχείρισης δικτύου (SNMP) ή οποιαδήποτε άλλη τυποποιημένη μέθοδο ερωτήματος.

Σήμερα, όλα τα UPS διαθέτουν θερμοκρασία αισθητήρα και εάν βρίσκεστε σε σύνθετη υποδομή, είναι πιθανό τα UPS να είναι παραγγελία και εκτός τυποποιημένου και, πιθανότατα, αυτός ο αισθητήρας μπορεί να ερευνηθεί μόνο με ένα εργαλείο που παρέχεται από τον πωλητή της UPS. Τώρα, η θερμοκρασία ενός UPS είναι μια κρίσιμη παράμετρος, ειδικά αν το UPS είναι ένα μεγάλο, προσαρμοσμένο UPS. Είναι πραγματικά είναι σημαντικό να παρακολουθείτε αυτές τις μετρήσεις. Φανταστείτε ότι το σύστημα ψύξης δεν λειτουργεί σωστά. λήψη σωστής ειδοποίησης όταν η θερμοκρασία φθάνει πάνω από το επίπεδο προειδοποίησης είναι θεμελιώδης. Απο την άλλη, προβλέποντας την αποτυχία θα εξοικονομήσει πολλά χρήματα. Επίσης, ακόμα και αν η φυσική η ζημιά δεν είναι πραγματικά ακριβή, ο χρόνος διακοπής μπορεί να κοστίσει πολλά χρήματα και να έχει ένα τρομερό αντίκτυπο στην επιχείρησή σας. Ένα καλό παράδειγμα είναι η περίπτωση μιας εμπορικής εταιρείας.

Σε αυτό το σενάριο, όλα πρέπει να είναι σε άριστη κατάσταση λειτουργίας. Σε αυτό το περιβάλλον, υπάρχει τρομερός ανταγωνισμός για την επίτευξη καλύτερων επιδόσεων έναντι των ανταγωνιστών - αγοράζοντας μια επιλογή μετοχών μερικά χιλιοστά του δευτερολέπτου πριν από τα άλλα είναι ένα μεγάλο πλεονέκτημα. Εδώ είναι είναι εύκολο να καταλάβουμε ότι, εάν οι διακομιστές δεν έχουν καλή απόδοση, είναι ήδη ένα ζήτημα. αν είναι κάτω, είναι μια πλήρης καταστροφή για την εταιρεία. Αυτό το παράδειγμα εξηγεί πώς είναι κρίσιμη η πρόβλεψη μιας αποτυχίας. Επιπλέον, είναι σημαντικό να καταλάβουμε πόσο κρίσιμο είναι αυτό είναι να ανακτήσετε όλες τις λειτουργικές παραμέτρους της υποδομής σας. Αυτό είναι όπου Zabbix έρχεται στη διάσωση, παρέχοντας ενδιαφέρουσες μεθόδους για την ανάκτηση δεδομένων που αλληλεπιδρούν με το λειτουργικό σύστημα, επιτρέποντάς σας να χρησιμοποιήσετε ένα εργαλείο γραμμής εντολών.

Οι απαντήσεις του Zabbix σε αυτό το είδος απαίτησης έχουν ως εξής:

- Εξωτερικοί έλεγχοι (πλευρά διακομιστή)
- UserParameter (πλευρά πράκτορα)
- Zabbix_sender: Αυτή η δυαδική μπορεί να χρησιμοποιηθεί και στην πλευρά του πράκτορα και του πλευρά του διακομιστή
- Ένα απλό, αποτελεσματικό και εύκολο στην εφαρμογή πρωτόκολλο επικοινωνίας

Το Zabbix παρέχει δυνατότητες για την κάλυψη όλων των στοιχείων που δεν μπορούν να ανακτηθούν με το πρότυπου πράκτορα. Στην πραγματική ζωή, είναι πιθανό ότι δεν μπορείτε να εγκαταστήσετε το πρότυπο πράκτορα στη συσκευή που θέλετε να παρακολουθήσετε. Ένα πρακτικό παράδειγμα είναι η UPS, όλοι οι διακομιστές που για κάποιο λόγο δεν μπορούν να διακυβευτούν κατά την εγκατάσταση εξωτερικών λογισμικό ή όλες τις συσκευές που δεν μπορούν να έχουν εγκατεστημένο προσαρμοσμένο λογισμικό.
Τώρα, για όλους αυτούς τους λόγους, δεν μπορείτε να έχετε έναν πράκτορα στη συσκευή σας, αλλά πρέπει να το κάνετε παρακολουθεί τις ζωτικές παραμέτρους αυτής της συσκευής, η μόνη εφικτή λύση για την οποία πρόκειται χρησιμοποιήστε έναν εξωτερικό έλεγχο.

Γενικοί κανόνες για τη γραφή σεναρίων

Αυτό το σενάριο καλύπτει όλα τα κρίσιμα σημεία που πρέπει να προσέξετε:

- Μην εισάγετε ανεπιθύμητους χαρακτήρες στην έξοδο
- Να γνωρίζετε τον τύπο. οπότε αν αναμένεται ένας αριθμός, αφαιρέστε όλα τα περιττάχαρακτήρες (όπως κεφαλίδες)
- Αποφύγετε τοπικές μετατροπές αριθμών. η περίπτωση της κουκκίδας και του κόμμα είναι ένα καλό παράδειγμα
- Έχετε ένα ημερολόγιο, έχοντας κατά νου ότι τα εξωτερικά σενάρια δεν είναι σειριοποιημένα, έτσι εσείς μπορεί να έχει τα μηνύματά σας μηνύματα στο αρχείο καταγραφής σας
- Έχετε υπόψη σας το χρόνο που δαπανάται από το σενάριο από το πότε ονομάζεται το σενάριο μέχρι το σενάριο παρέχει την έξοδο

Αυτά τα σενάρια, βεβαίως, τρέχουν με το χρήστη του διακομιστήZabbix, έτσι ίσως εσύ πρέπει να φροντίζουν τα δικαιώματα αρχείων και τα δικαιώματα sudo.
Θυμηθείτε ότι, εάν το ζητούμενο σενάριο δεν βρέθηκε ή ο διακομιστήςZabbix δεν έχει δικαιώματα εκτέλεσης, το στοιχείο δεν θα υποστηρίζεται. Επίσης, στην περίπτωση α timeout, και στις δύο προηγούμενες περιπτώσεις θα εμφανιστεί ένα μήνυμα σφάλματος και το διχαλωτή διαδικασία για το σενάριο θα σκοτωθεί.

Επέκταση του Zabbix

Η κατανόηση του πρωτοκόλλου παρακολούθησης Zabbix μας επιτρέπει να γράφουμε σενάρια, agents και προσαρμοσμένους ανιχνευτές. Με άλλα λόγια, μας επιτρέπει να επεκτείνουμε ελεύθερα την παρακολούθηση του Zabbix με την επέκταση των μέσων συλλογής δεδομένων.

Ωστόσο, το μόνο σημείο πρόσβασης που έχουμε αναφέρει έως τώρα είναι το γραφικό περιβάλλον. Είτε θέλετε να προσθέσετε ένα χρήστη, να αλλάξτε τη συχνότητα δειγματοληψίας ενός item ή να εξετάσετε ιστορικά δεδομένα, πρέπει πάντα να χρησιμοποιείτε το γραφικό περιβάλλον. Αυτή είναι σίγουρα μια βολική λύση για καθημερινές δραστηριότητες, όπως το μόνο που χρειάζεστε έχει πρόσβαση σε ένα απλό πρόγραμμα περιήγησης.

Από την άλλη πλευρά, δεν μπορεί να εκτελεστεί κάθε μεγάλη και πολύπλοκη λειτουργία βολικά μέσω μιας διαδικτυακής εφαρμογής, και μερικές φορές, δεν χρειάζεται να απεικονίζετε τα δεδομένα, αλλά πρέπει να τα εξάγετε και να τα τροφοδοτήσετε σε άλλα προγράμματα για να τα αναλύσετε περαιτέρω.

Εξερεύνηση του API Zabbix

Το Zabbix παρέχει ένα σημείο εισόδου για αλληλεπίδραση, χειρισμό, διαμόρφωση και δημιουργία items στο Zabbix. Το API διατίθεται μέσω του PHP frontend στο http: // <your-zabbix-server> /zabbix/api_jsonrpc.php. Το πρωτόκολλο επικοινωνίας βασίζεται σε JSON και το μέσο που χρησιμοποιείται είναι προφανές HTTP / HTTPS.

Το API JSON-RPC του Zabbix παρέχει μια ωραία διεπαφή και διαθέτει πολλές λειτουργίες.

Μόλις γίνει το authentication και πιστοσποιηθεί η σύνδεση, θα σας επιτρέψει να πραγματοποιήσετε οποιαδήποτε λειτουργία στο Zabbix. Το API του Zabbix μπορεί να είναι πραγματικά χρήσιμο. Ως πρακτικό παράδειγμα, μπορείτε να θεωρήσετε ότι εσείς ενδεχομένως να χρειαστείτε να προσθέσετε έναν μεγάλο αριθμό συσκευών οι οποίες, πιθανότατα, έχουν ήδη οριστεί σε ξεχωριστό αρχείο. Το API παρέχει το σημείο εισόδου για να τα προσθέσετε όλα στο Zabbixserver χρησιμοποιώντας απλά ένα ειδικό σενάριο.

Το API Zabbix εισήχθη με την έκδοση Zabbix 1.8 και πέρασε τις αλλαγές μέχρι την τρέχουσα έκδοση 2.4. Αυτή η έκδοση μπορεί να θεωρηθεί πιο σταθερή και ώριμη, αλλά εξακολοθεί να

βελτιώνεται, ώστε τα πράγματα να αλλάξουν λίγο τις μελλοντικές εκδόσεις. Αυτό δεν σημαίνει ότι δεν είναι κατάλληλο για παραγωγή περιβάλλον/ Αντίθετα, όσο μεγαλύτερη είναι η εγκατάσταση, τόσο πιο συχνή μπορεί να είναι τη χρήση του API σε δέσμες ενεργειών για πολύπλοκες και χρονοβόρες διαδικασίες.

Ο παρακάτω κώδικας είναι ένα απλοποιημένο αίτημα JSON στο API του Zabbix:

```
{
}}
"jsonrpc": "2.0",
"method": "method.name",
"parameters": {
"param_1_name": "param_1_value",
"param_2_name": "param_2_value"
},
"id": 1,
"auth": "159121ba47d19a9b4b55124eab31f2b81"
```

Τα ακόλουθα σημεία εξηγούν τι αντιπροσωπεύουν οι προηγούμενες γραμμές κώδικα:

- "jsonrpc": "2.0": Αυτή είναι μια τυπική παράμετρος JSON PRC που χρησιμοποιείται για ναπροσδιορίσει την έκδοση πρωτοκόλλου. αυτό δεν θα αλλάξει στα αιτήματά σας.
- "method": "method.name": Αυτή η παράμετρος ορίζει τη λειτουργία που πρέπει να εκτελείται. για παράδειγμα, μπορεί να είναι host.create ή item.update.

Συμπέρασμα

Το Zabbix είναι αν όχι το κορυφαίο, ένα από τα καλύτερα ολοκληρωμένα εργαλεία παρακολούθησης. Με τον όρο ολοκληρωμένα, εννοούμε πως υποστηρίζει όλων των ειδών ελέγχων, αναπτυγμένο σύστημα ειδοποιήσεων, και απεικονισης καθώς πολλά προγράμματα βγάζουν plugins γι αυτό. Διαθέτει API με το οποίο μπορείτε να παρακολουθήσετε οτιδήποτε θέλετε τόσο σε επίπεδο υποδομών, όσο και σε επίπεδο προγραμμάτων. Είναι ανοιχτού κώδικα, παρέχεται δωρεάν και αναπτύσεταιαπο ένα τεράστιο community. Στην τελευταία έκδοση του υποστηρίζει ενσωμάτωση με βάσεις δεδομένων χρονοσειρών(TimescaleDB, Prometheus). Το Zabbix μας βοηθάει να προβλέψουμε και να μεγαλώσουμε την συστημική υποδομή και το δίκτο μας, να αυτοματοποιήσουμε τις εσωτερικές διεργασίες, και πιο απλά, να βρίσκουμε τυχόν σφάλματα, να εξοικονωμούμε χρόνο και χρήματα.

Βιβλιογραφία

https://en.wikipedia.org/wiki/System_monitor

https://en.wikipedia.org/wiki/Network_monitoring

https://en.wikipedia.org/wiki/Zabbix

https://www.zabbix.com/documentation/4.0/manual/

The Art of Monitoring Book - James TurnBull

Unix and Linux System Administration Handbook ISBN-13: 978-0134277554

Mastering Zabbix - Second Edition ISBN-13: 978-1785289262

Zabbix Network Monitoring Essentials ISBN-13: 978-1784399764

https://www.talentica.com/assets/tech-watch/techwatch-report-monitoring-tools.pdf

Time Series Database (TSDB) Explained | InfluxData