

Το Υπολογιστικό Νέφος
και
το οικοσύστημα ανάπτυξης λογισμικού

Πτυχιακή εργασία

Μιχόπουλος Χαράλαμπος ΜΕ/09062

Πανεπιστήμιο Πειραιώς

Τμήμα Ψηφιακών Συστημάτων

ΠΜΣ: Δικτυοκεντρικά Συστήματα

Αντί προλόγου - Σκοπός

Ο όρος utility computing είχε ακουστεί από τη δεκαετία του 1960 και αναφερόταν στη διαθεσιμότητα υπολογιστικών πόρων με τη μορφή δημοσίων αγαθών όπως το ηλεκτρικό ρεύμα ή η τηλεφωνία. Εν τούτοις, μόλις πρόσφατα η ιδέα αυτή άρχισε να παίρνει σάρκα και οστά και να υλοποιείται υπό τον όρο cloud computing. Από το 2006 όταν παρουσιάστηκε η πρώτη ολοκληρωμένη υπηρεσία cloud computing έχουν παρουσιαστεί πολλές μορφές υπηρεσιών cloud. Επίσης, πριν από αυτό υπήρχαν παρεμφερής υπηρεσίες, προάγγελοι του cloud computing. [1]

Πλέον, το πεδίο αυτό αναπτύσσεται με ταχύτατους ρυθμούς. Δεν είναι υπερβολή να πούμε ότι καθημερινά εμφανίζονται νέες υπηρεσίες cloud computing, νέες μορφές υπηρεσιών και νέες ιδέες. Αν και το τοπίο δεν έχει ακόμη ξεκαθαρίσει - δεν υπάρχει καν ένας σαφής και κοινά αποδεκτός ορισμός του τι είναι cloud computing - τα πάντα κινούνται γύρω από αυτό, επηρεάζοντάς το με ασύλληπτους ρυθμούς.

Στόχος αυτής της πτυχιακής δεν είναι να εμβαθύνει σε πολύ συγκεκριμένα χαρακτηριστικά του cloud computing αφού το πεδίο είναι ακόμα μεταβαλλόμενο και μάλιστα με ταχείς ρυθμούς. Χαρακτηριστικό αυτής της ταχύτητας είναι πως αρκετά τμήματα αυτής της διατριβής αλλάχθηκαν, αφού οι εξελίξεις τα ξεπερνούσαν. Έτσι, ο σκοπός μεταβλήθηκε προς πιο γενικά ζητήματα, όπως το να παρουσιάσει με όσο πιο απλό τρόπο γίνεται το τι είναι cloud computing, να αναλύσει τα κίνητρα που το ανέπτυξαν, να παρουσιάσει τα προβλήματα που μπορεί να λύσει αλλά και τα νέα ζητήματα που εγείρει καθώς και να κάνει μια σύντομη ανασκόπηση των λύσεων και των υπηρεσιών που προσφέρονται. Ο τελικός στόχος δεν είναι να αποτελέσει οδηγίες χρήσης για μια συγκεκριμένη υπηρεσία ή έναν οδηγό για την εφαρμογή του cloud computing σε κάποιον οργανισμό. Κάτι τέτοιο άλλωστε ούτε υπάρχει ούτε θα υπάρξει.

Όπως θα δούμε το cloud computing δεν είναι πανάκεια ούτε μια απλή λύση τύπου plug-n-play. Αντιθέτως, απαιτεί σωστή αντίληψη του τι προσφέρει και τι απαιτεί, ορθή αξιολόγηση των αναγκών κάθε οργανισμού και ουσιαστική μελέτη πριν την εφαρμογή του. Μέσα από το πρώτο μέρος αυτού του συγγράμματος, ο αναγνώστης θα έχει την ευκαιρία να έρθει σε επαφή με την κεντρική ιδέα του cloud computing, να πάρει μια εικόνα τόσο για τη σημερινή κατάσταση στο θέμα όσο και για το πού οδεύει και να γνωρίσει το τι μπορεί να προσφέρει.

Το δεύτερο μέρος του συγγράμματος ασχολείται με πιο πρακτικά ζητήματα και επικεντρώνεται στο ρόλο που μπορεί να διαδραματίσει η νέα αυτή τεχνολογία στην ανάπτυξη λογισμικού. Αν και ο βασικός λόγος υιοθέτησης αυτής της τεχνολογίας, όπως θα δούμε, έχει να κάνει περισσότερο με οικονομικά και διαχειριστικά ζητήματα, μπορεί να υποβοηθήσει και να επιταχύνει το software development, μειώνοντας παράλληλα το κόστος.

Πλέον, η ανάπτυξη λογισμικού έχει εξελιχθεί σε μια αρκετά πολύπλοκη διαδικασία, με αυξημένες απαιτήσεις και την εμπλοκή αρκετών ομάδων. Αυτό έχει προφανώς επιφέρει και αύξηση του κόστους και αύξηση της πολυπλοκότητας στη διαχείριση των διαφόρων περιβαλλόντων που απαιτούνται. Αφού παρουσιάσουμε τις κυριότερες υπηρεσίες cloud computing, θα εξετάσουμε το πώς μπορούν να αξιοποιηθούν για την βελτίωση του κύκλου της ανάπτυξης και πώς μπορούν να επηρεάσουν το οικοσύστημα ανάπτυξης. Θα δούμε ότι υπάρχουν ήδη προτάσεις που περιλαμβάνουν αρκετά συστήματα που εμπλέκονται στην ανάπτυξη και των οποίων η χρήση πλέον θεωρείται αυτονόητη. Συνεχίζοντας θα παρουσιάσουμε τις λύσεις που μπορεί να προσφέρει το cloud computing σε όλες τις ομάδες συγγραφής κώδικα και ελέγχου πάνω σε ρεαλιστικά σενάρια.

Εισαγωγή στο Cloud Computing

Εισαγωγή

Σήμερα, ακούγεται εντονότατα ο όρος “Cloud computing”. Όλα τα μέσα πληροφορικής παρουσιάζουν άρθρα για αυτό και καθημερινά εμφανίζονται νέες λύσεις για το “Cloud”. Παρότι ακόμα υπάρχουν διενέξεις και διαφωνίες για το τι ακριβώς είναι - ή πρέπει να είναι - το “Cloud”, το “Cloud Computing” είναι γεγονός και η εποχή του είναι μπροστά μας .

Θα προσπαθήσουμε να αποσαφηνίσουμε τον όρο αυτό, να παρουσιάσουμε τους τομείς που μπορεί να επηρεάσει και φυσικά να προσδιορίσουμε το ρόλο που μπορεί να συντελέσει στην ανάπτυξη λογισμικού.

Το cloud computing μοιάζει να είναι κάτι περισσότερο από μια υπηρεσία που προσφέρει ένα data center ή μια server farm. Είναι μάλλον ένα σύνολο εξελιγμένων τεχνολογιών που επιτρέπει στους οργανισμούς να διαχειρίζονται καλύτερα τους πόρους τους. Αυτή η προσέγγιση δείχνει ότι το cloud computing είναι περισσότερο ενά επιχειρησιακό-διαχειριστικό μοντέλο παρά ένα τεχνολογικό μοντέλο. Σήμερα, το λογισμικό έχει καταστεί αναπόσπαστο κομμάτι των επιχειρησιακών διαδικασιών και δεν νοείται εταιρία χωρίς μηχανογράφηση. Αυτό βέβαια σημαίνει ότι το λογισμικό θα πρέπει να είναι σε θέση να ακολουθήσει ή και να συμβάλει στην εξέλιξη της εταιρίας.

Η έντονη ανάπτυξη λογισμικού όμως, έχει και τα μειονεκτήματά της. Κάθε εταιρία έχει πολλά υπολογιστικά συστήματα με διαφορετικές εφαρμογές και η διαχείριση (administration) αυτών έχει καταστεί πολύπλοκη, πολυέξοδη και χρονοβόρα. Το cloud computing έρχεται με κύριο στόχο να βελτιώσει αυτά τα ζητήματα καθιστώντας τη διαχείριση πιο ευέλικτη, με μικρότερες απαιτήσεις σε επιχειρηματικούς πόρους.

Βέβαια, το cloud computing δεν είναι μια γρήγορη λύση που μπορεί απλώς άμεσα να εφαρμοστεί και να λύσει ως δια μαγείας τα προβλήματα διαχείρισης. Αντίθετα, απαιτεί έρευνα από τον καθέναν που θα το χρησιμοποιήσει, ούτως ώστε να επιλεγεί η ορθή προσέγγιση. Άλλωστε, όπως κάθε νέα τεχνολογία, γεννάει και αυτό νέα προβλήματα και είναι απαραίτητη η σωστή αποτίμηση των νέων προβλημάτων που εισάγονται και των παλαιών που αντιμετωπίζονται, ώστε η λύση που θα επιλεγεί να είναι συμφέρουσα.

Το cloud computing, λοιπόν, απαιτεί αυτός που θα το εφαρμόσει να έχει ήδη μια ισχυρή βάση από βέλτιστες πρακτικές στον τομέα τόσο της ανάπτυξης όσο και της διαχείρισης λογισμικού.

Στα επόμενα κεφάλαια θα παρουσιάσουμε το περιβάλλον του cloud μαζί με τις λύσεις που προτείνονται και τα προβλήματα που ανακύπτουν.

Ορολογία

Πριν προχωρήσουμε στα ενδότερα του cloud computing είναι χρήσιμο να αποσαφηνίσουμε μερικούς όρους που χρησιμοποιούνται στο cloud computing.

Virtualization

Είναι η τεχνική δημιουργίας σε επίπεδο λογισμικού κάτι εικονικού, όπως υλικό, δίκτυο κλπ.

Virtual Machines

Είναι εικονικοί υπολογιστές οι οποίοι “τρέχουν” σε κανονικούς. Από την οπτική γωνία του λειτουργικού συστήματος καθώς και αυτή του τελικού χρήστη δεν έχουν διαφορά από έναν κανονικό υπολογιστή. Από την οπτική γωνία του διαχειριστή η διαφορά έγκειται στο ότι απαιτείται ειδικό λογισμικό που θα προσφέρει το περιβάλλον εκτέλεσης που θα αποτελεί το ενδιάμεσο ανάμεσα στην εικονική μηχανή και το υλικό, αφού η εικονική μηχανή δεν έχει άμεση πρόσβαση σε αυτό.

Hypervisor

Ο hypervisor ή διαχειριστής εικονικών μηχανών (Virtual Machine Manager, VMM) είναι μια τεχνική hardware virtualization που επιτρέπει την εκτέλεση πολλαπλών λειτουργικών συστημάτων, που ονομάζονται *guests* σε έναν υπολογιστή ο οποίος λέγεται *host*. Προσφέρει στο φιλοξενούμενο λειτουργικό σύστημα μια εικονική πλατφόρμα και ελέγχει την εκτέλεση των εικονικών μηχανών.

Virtualization Extensions

Είναι επεκτάσεις του instruction set των επεξεργαστών για την εγγενή (σε επίπεδο hardware δηλαδή) υποστήριξη του virtualization. Πρόκειται για μια προσθήκη με εξαιρετική σημασία αφού προσφέρει ιδιαίτερη αύξηση στις επιδόσεις των εικονικών μηχανών. [2]

Τι είναι το cloud computing

Όλοι έχουν έναν διαφορετικό ορισμό για τον όρο “cloud computing”. Μπορεί να είναι η δυνατότητα ενοικίασης ενός έως χιλιάδες servers, σε διάφορα γεωγραφικά σημεία, ώστε μια εφαρμογή να εκτελείται σε πολλά και ισχυρά συστήματα. Μπορεί να είναι η ενοικίαση ενός virtual server, η εγκατάσταση λογισμικού σε αυτόν, η ενεργοποίηση και απενεργοποίησή του κατά το δοκούν και η κλωνοποίησή του σε δεκάδες κλώνους ώστε να αντιμετωπιστεί μια ξαφνική αύξηση της χρήσης της εφαρμογής. Μπορεί να είναι η αποθήκευση αμέτρητων δεδομένων και η χρήση τους από εξουσιοδοτημένους χρήστες. Μπορεί να υποστηρίζεται από έναν πάροχο υπηρεσιών cloud που προσφέρει μια πλατφόρμα η οποία περιλαμβάνει web servers, database servers, perl, python, php και τη δυνατότητα όλα αυτά να κλιμακώνονται άμεσα. Μπορεί να είναι απλώς η δυνατότητα να χρησιμοποιούμε εφαρμογές στο διαδίκτυο, από απλή αποστολή και λήψη email, μέχρι εφαρμογές πωλήσεων. Μπορεί να είναι ένας αποθηκευτικός χώρος για προσωπικά δεδομένα, όπως αρχεία, φωτογραφίες κλπ. Τέλος, μπορεί να είναι η δυνατότητα να

αξιοποιούμε πολλές web services για να παρέχουμε σε μια ιστοσελίδα ενοποιημένα δεδομένα και υπηρεσίες από διάφορες πηγές. [3]

Δεδομένης της ασάφειας γύρω από το cloud computing καθώς και των πολλών διαστάσεών τους, μπορούμε να ισχυριστούμε ότι ο εν λόγω όρος εμπεριέχει όλα τα παραπάνω. Σύμφωνα με το eucalyptus, ένα open source cloud system, το cloud είναι η πρόσβαση σε υπολογιστές και τη λειτουργικότητα που παρέχουν μέσω του διαδικτύου ή ενός τοπικού δικτύου. Οι χρήστες του cloud αιτούνται την πρόσβαση σε ένα σύνολο δικτυακών υπηρεσιών οι οποίες διαχειρίζονται υπολογιστικούς πόρους (π.χ. μηχανήματα, δίκτυο, αποθηκευτικούς χώρους, λειτουργικά συστήματα, περιβάλλοντα ανάπτυξης εφαρμογών, εφαρμογές). Με την αποδοχή της αίτησης εκχωρούνται στο χρήστη αποκλειστικά οι απαιτούμενοι πόροι οι οποίοι επιστρέφονται στο cloud όταν ολοκληρωθεί η εργασία. Ονομάζεται "cloud computing" επειδή ο χρήστης δεν μπορεί να δει ή να ορίσει τη φυσική τοποθεσία και την οργάνωση των υπολογιστικών συστημάτων που του παρέχουν τους απαιτούμενους πόρους. Με άλλα λόγια, οι αιτούμενοι πόροι εξασφαλίζονται από ένα σύνολο (ή σύννεφο) πόρων στο οποίο επιστρέφονται όταν δεν χρειάζονται πλέον. Το σύνολο - ή σύννεφο - αυτό αποτελείται από μηχανές και δικτυακές υπηρεσίες που υλοποιούν το cloud computing. [4]

Όπως θα δούμε, υπάρχουν διάφορα είδη clouds, με διαφορετικές ταξινομήσεις, ενώ διαχωρισμός υπάρχει και στο είδος των υπηρεσιών που κάθε cloud δύναται να παρέχει. Για να καλύψουμε λοιπόν την έννοια "cloud computing" αφ' ενός θα αναφερθούμε στα κοινά χαρακτηριστικά που πρέπει να έχουν οι υποδομές cloud και κατόπιν θα εξετάσουμε πιο λεπτομερώς τα διάφορα είδη και την ταξινόμησή τους.

Ιστορικά στοιχεία

Ο όρος "cloud" χρησιμοποιείται μεταφορικά για το διαδίκτυο δανεισμένος από παλαιότερα διαγράμματα που χρησιμοποιούσαν ένα "συννεφάκι" για να αναπαραστήσουν το τηλεφωνικό δίκτυο και αργότερα, σε διαγράμματα υπολογιστών για την αναπαράσταση δικτύων υπολογιστών, υπονοώντας πάντα μια "αφαίρεση" (abstraction) των εσωτερικών του εκάστοτε δικτύου.

Το "cloud computing" θεωρείται η φυσική συνέχεια της ευρείας αποδοχής τεχνολογιών όπως το virtualization, η αρχιτεκτονική SOA (Service-Oriented Architecture) και του "utility computing". Σε αυτές τις τεχνολογίες, αφαιρούνται τα εσωτερικά χαρακτηριστικά των συστημάτων από τους τελικούς χρήστες καθώς και η ανάγκη οι τελευταίοι να έχουν πόρους και τεχνογνωσία για την υποστήριξή τους.

Η κεντρική ιδέα πίσω από το "cloud computing" απαντάται πίσω στη δεκατία του 1960, όταν ο John McCarthy είχε δηλώσει πως "ίσως, κάποια μέρα, τα υπολογιστικά συστήματα να είναι οργανωμένα και να διατίθενται ως δημόσια αγαθά". Τα βασικά χαρακτηριστικά του "cloud computing" και η ομοιότητά του με άλλα δημόσια αγαθά όπως το ρεύμα ή το τηλέφωνο,

αναλύθηκαν σε ένα βιβλίο του 1966 με τίτλο “The challenge of computer utility” του Douglas Parkhill.

Ο όρος cloud, δανεισμένος από τα τηλεφωνικά δίκτυα, μέχρι τις αρχές της δεκαετίας του 1990 υπονοούσε κυκλώματα από το ένα άκρο στο άλλο, ενώ αργότερα εικονικά ιδιωτικά δίκτυα (VPNs) όπου μέρος των δικτύων παρεχόταν σε συγκεκριμένους χρήστες. Η ιδέα αυτή πέρασε στους servers και στη γενικότερη υποδομή των δικτύων. [1]

Μετά την έκρηξη του web, η Amazon ήταν αυτή που έπαιξε καθοριστικό ρόλο στην εξέλιξη του cloud computing. Όπως και στις περισσότερες μεγάλες εταιρίες, οι υπολογιστικοί πόροι ήταν άφθονοι, με μέση χρήση κάτω του 10%, απλώς μόνο και μόνο για να μπορούν να ανταπεξέλθουν σε ξαφνικές εκρήξεις του φόρτου εργασίας και της ζήτησης. Θρυλείται πως το έναυσμα για την υλοποίηση του cloud τους ήταν αυτή ακριβώς η χαμηλή χρήση των υπολογιστών τους και η πίστη πως η ενοικίαση των μη χρησιμοποιούμενων πόρων θα μπορούσε να είναι κερδοφόρος. Έτσι, το 2006 παρουσίασαν τα Amazon Web Services (AWS) στη βάση του computing utility.

Στις αρχές του 2008, το eucaalyptus έγινε το πρώτο σύστημα ανοιχτού λογισμικού που προσέφερε μια πλατφόρμα συμβατή με αυτή της Amazon. Την ίδια εποχή, εμφανίστηκε και το OpenNebula βελτιωμένο από ένα έργο της ευρωπαϊκής ένωσης, το RESERVOIR, και αποτελούσε το πρώτο σύστημα ανοιχτού κώδικα που προσέφερε τη δυνατότητα για private και hybrid clouds. Από εκεί και πέρα, η εξέλιξη είναι ραγδαία, με πάρα πολλές εταιρίες να προσφέρουν υπηρεσίες cloud computing σε διάφορα επίπεδα.

Χαρακτηριστικά μιας υποδομής cloud

Ελαστικότητα και κλιμάκωση

Οι δύο αυτοί όροι αναφέρονται στην ελαστικότητα του cloud και κατ'επέκταση στη δυνατότητα κλιμάκωσής του. Με τον όρο ελαστικότητα (elasticity) εννοούμε ότι το cloud, από τη σκοπιά του χρήστη, είναι ελαστικό. Φυσικά αναφερόμαστε στο χρήστη του cloud και όχι στο χρήστη μιας εφαρμογής που βρίσκεται σε αυτό. Δηλαδή, το cloud θα πρέπει να είναι σε θέση να παρέχει περισσότερους πόρους όταν η εφαρμογή τους απαιτεί και φυσικά να τους δέχεται πίσω όταν δεν χρειάζονται.

Επομένως, κάθε εφαρμογή στο cloud θα πρέπει να μπορεί να κλιμακώνεται (scalability) σύμφωνα με τις απαιτήσεις του φόρτου κάθε στιγμή χωρίς να χρειάζεται η επέμβαση διαχειριστών (administrators) και η εγκατάσταση νέων συστημάτων γι' αυτό. Αντίθετα, το cloud είναι αυτό που είναι υπεύθυνο για την παροχή των απαιτούμενων πόρων. [5]

Αυτο-τροφοδοσία (Self-service provisioning)

Με τον όρο αυτό εννοούμε ότι ο χρήστης θα πρέπει να είναι σε θέση να λαμβάνει τους πόρους που απαιτεί από το cloud χωρίς χρονοβόρες διαδικασίες. Ας το αντιπαραθέσουμε με ένα κλασσικό data-center. Εκεί πρέπει να “στηθεί” - ή να αγοραστεί αν δεν υπάρχει - ένας νέος server, να ρυθμιστεί και μετά να γίνει η εγκατάσταση της εφαρμογής ώστε να αρχίσει να τρέχει. Στον κόσμο του cloud, αυτό δεν είναι αποδεκτό. Εάν απαιτείται άμεσα ένας ακόμα server, θα πρέπει να είναι άμεσα διαθέσιμος.

Το χαρακτηριστικό αυτό δίνει άσφρα με το προηγούμενο, αφού οι χρονοβόρες διαδικασίες για την παροχή νέων πόρων είναι τροχοπέδη για την άμεση κλιμάκωση. Κάτι, φυσικά που προσφέρει η ελαστικότητα. [5]

Application Programming Interfaces (API)

Καταρχήν οι παρόχοι υπηρεσιών cloud θα πρέπει να προσφέρουν ένα API ώστε οι εφαρμογές στο cloud να μπορούν να επικοινωνούν με την υποδομή. Αυτό είναι απαραίτητο για τις ίδιες τις εφαρμογές, αφ’ ενός για να μπορούν να αξιοποιήσουν υπηρεσίες του cloud και αφ’ εταίρου για να μπορούν και οι ίδιες να αυξομειώσουν μόνες τους το μέγεθος τους χωρίς την ανθρώπινη παρέμβαση. Αυτό τους επιτρέπει να είναι αυτόνομες, πιο αξιόπιστες, να διαχειρίζονται καλύτερα τις οριακές συνθήκες πχ ξαφνικός φόρτος και φυσικά συμβάλλει στη μείωση του κόστους.

Βέβαια, όπως έχει αποδειχτεί πάμπολες φορές στο παρελθόν και σε διάφορους τομείς της πληροφορικής, είναι απαραίτητο, τα APIs αυτά να είναι κοινά και standardized - προαπαιτούμενο και για το επόμενο χαρακτηριστικό. [5]

No vendor lock-in

Θεωρείται σχεδόν αυτονόητο πλέον πως ένας χρήστης μιας υπηρεσίας cloud δεν θα πρέπει να κλειδώνεται στον κάθε πάροχο. Αντίθετα, θα πρέπει να είναι σε θέση να μεταβεί σε άλλη πλατφόρμα άμεσα. Μπορούμε πολύ εύκολα να φανταστούμε το πρόβλημα ενός χρήστη εάν ο πάροχος δεν προσφέρει τις αναμενόμενες υπηρεσίες ή αυξήσει πολύ τις τιμές του, ενώ έχει ήδη κλειδώσει με το κλειστό του API τον πελάτη. Τότε, το κόστος για τον τελευταίο να μεταβεί σε νέα πλατφόρμα θα είναι υψηλό. [5]

Μέτρηση και χρέωση υπηρεσιών

Κάθε πάροχος υπηρεσιών cloud θα πρέπει να προσφέρει, εκτός από τις υπηρεσίες του και εργαλεία μέτρησης των πόρων που χρησιμοποιεί ο κάθε πελάτης και φυσικά ακριβείς και άμεσες πληροφορίες σχετικά με τις χρεώσεις. Σε πολλές περιπτώσεις μάλιστα, είναι απαραίτητο και για τον πελάτη να έχει αντίστοιχα συστήματα ώστε να υπολογίζει και ο ίδιος το κόστος χρήσης και να το συγκρίνει με αυτό που δίνει ο πάροχος. Έτσι, έχουμε τόσο την απαίτηση μετρήσεων από πλευράς παρόχου όσο και απαιτήσεις για μετρήσεις από πλευράς χρήστη. [5]

Παρακολούθηση απόδοσης

Κάθε πάροχος cloud θα πρέπει να προσφέρει και ένα σύστημα διαχείρισης (management) του περιβάλλοντος στον πελάτη. Το σύστημα αυτό πρέπει να δίνει τη δυνατότητα παρακολούθησης τόσο των φυσικών υπολογιστικών συστημάτων όσο και των εφαρμογών. Έτσι σε περίπτωση που υπάρχει κάποιο SLA (Service Level Agreement) θα μπορούν και οι δύο πλευρές να παρακολουθούν πόσο αυτό τηρείται.

Το σύστημα ελέγχου θα πρέπει να παρέχει άμεσα πληροφορίες για σημεία-κλειδιά του συστήματος, όπως ταχύτητα απόκρισης, χρησιμοποιούμενοι πόροι κλπ.

Πολλές υπηρεσίες cloud προσφέρουν ένα dashboard από το οποίο ο χρήστης παρακολουθεί αυτά τα σημεία κλειδιά καθώς και συστήματα monitoring διάφορων χαρακτηριστικών. [5]

Ασφάλεια

Πολλοί θεωρούν πως ένα cloud περιβάλλον είναι ασφαλές για τις εφαρμογές τους, βασιζόμενοι στην αξιοπιστία του παρόχου. Εν τούτοις, ο καθένας θα πρέπει να διασφαλίζει πως τα δεδομένα που μεταφέρονται στο cloud παραμένουν ασφαλή και μακριά τόσο από τους υπόλοιπους χρήστες του ίδιου cloud όσο και από πιθανούς εισβολείς (hackers). Φυσικά, τα θέματα ασφαλείας της εφαρμογής παραμένουν και θα πρέπει να αντιμετωπίζονται όπως και πριν.

Πολλές εταιρίες, λόγω της φύσης της δουλειάς τους, πρέπει να συμμορφώνονται με πολύπλοκους κανόνες ασφαλείας αναφορικά με τα δεδομένα τους. Εάν ο πάροχος του cloud δεν είναι σε θέση να εξασφαλίσει ένα σημαντικό επίπεδο ασφαλείας, τότε σίγουρα και οι εταιρίες αυτές δεν θα μπορούν να πιστοποιήσουν τη συμμόρφωσή τους με τους κανόνες ασφαλείας.

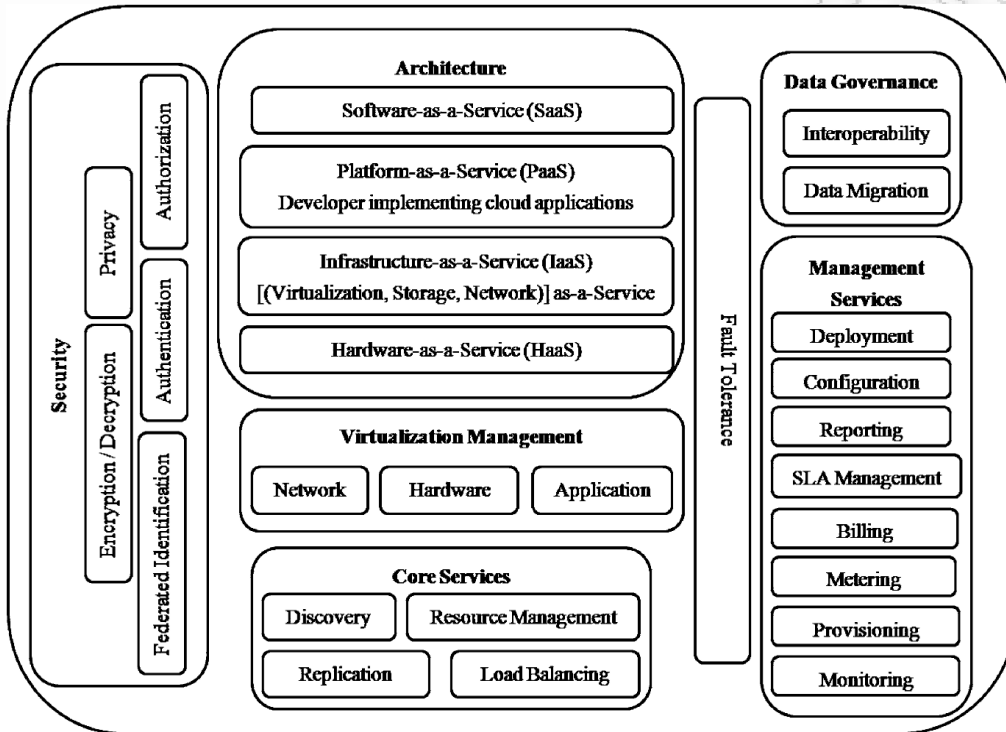
Επίσης, πρέπει να αναφερθεί και το ζήτημα της γεωγραφικής θέσης αποθήκευσης των δεδομένων. Πολλές χώρες έχουν αυστηρούς κανόνες ασφαλείας για ευαίσθητα προσωπικά δεδομένα και δεν επιτρέπουν την έξοδο αυτών από τα όρια της χώρας. Στις περιπτώσεις αυτές θα πρέπει να λαμβάνονται υπόψιν και γεωγραφικά κριτήρια για τη χρήση του cloud. [5]

Αρχιτεκτονική του Cloud

Με τον όρο αρχιτεκτονική του cloud εννοούμε το σχεδιασμό εφαρμογών λογισμικού οι οποίες χρησιμοποιούν κατά απαίτηση πόρους διαθέσιμους στο διαδίκτυο. Η αρχιτεκτονική αυτή βασίζεται σε συστήματα τα οποία λαμβάνουν υπολογιστικούς πόρους - ως υπηρεσίες - όταν τους χρειάζονται, επιτελούν την εργασία τους και έπειτα τους επιστρέφουν στο cloud. Οι

υπηρεσίες είναι διαθέσιμες οπουδήποτε, αλλά το σύστημα του cloud εμφανίζεται ως το σημείο επαφής των εξωτερικών συστημάτων με τους πόρους που χρειάζονται.

Παρακάτω βλέπουμε τα βασικά χαρακτηριστικά ενός συστήματος cloud. [6]



Ταξινόμηση μοντέλων παροχής υπηρεσιών Cloud

Γενικά υπάρχουν τρεις μορφές με τις οποίες προσφέρονται οι υπηρεσίες cloud computing. Αυτές είναι Infrastructure as a Service, Platform as a Service και Software as a Service. Κάθε μια από αυτές καλύπτει συγκεκριμένο φάσμα αναγκών.

Στο Infrastructure as a Service επίπεδο προσφέρονται υπηρεσίες αποθήκευσης (storage) και υπολογιστικών πόρων που μπορούν να αξιοποιηθούν από τους προγραμματιστές και το IT ώστε να καλύψουν επιχειρησιακές απαιτήσεις

Στο επίπεδο του Platform as a Service προσφέρονται περιβάλλοντα ανάπτυξης τα οποία αξιοποιούνται για την ανάπτυξη και εκτέλεση των εφαρμογών.

Τέλος, στο επίπεδο του Software as a Service προσφέρονται έτοιμες λύσεις που μπορούν είτε να χρησιμοποιηθούν από τον τελικό χρήστη είτε να ενσωματωθούν σε κάποια εφαρμογή μέσω web services ή παρόμοιων τεχνολογιών.

Στη συνέχεια θα αναλύσουμε περαιτέρω κάθε επίπεδο.

Infrastructure as a Service

Οι υπηρεσίες υποδομής (Infrastructure as a Service, IaaS) αφορούν την παροχή υπολογιστικών πόρων στο χαμηλότερο επίπεδο, όπως servers, storage, networking technology, ως υπηρεσίες. Επίσης, μπορεί να περιλαμβάνουν και την παροχή τεχνολογιών virtualization καθώς και λειτουργικών συστημάτων (operating systems).

Ο χρήστης-πελάτης μιας υπηρεσίας IaaS μπορεί, αντί να αγοράσει έναν server και το αντίστοιχο λειτουργικό σύστημα να ενοικιάσει έναν virtual server με εγκατεστημένο το λειτουργικό σύστημα της επιλογής του και να χρεώνεται με βάση τη χρήση που κάνει.

Η κλιμάκωση (scaling) σε αυτό το επίπεδο έχει να κάνει με την ενοικίαση περισσότερων virtual servers ή την χρήση ενός με μεγαλύτερες δυνατότητες. Δηλαδή, εκτός από horizontal scaling προσφέρεται και vertical scaling, αφού ο χρήστης μπορεί άμεσα να εκκινήσει ένα virtual machine με καλύτερα χαρακτηριστικά, πχ περισσότερη μνήμη ή περισσότερους πυρήνες επεξεργαστή, χωρίς φυσικά να πρέπει να αγοράσει αντίστοιχο μηχάνημα, ούτε να εγκαταστήσει ξανά το λειτουργικό σύστημα.

Σήμερα, ο πιο δημοφιλής πάροχος υπηρεσιών υποδομής στο cloud είναι το Amazon. Η αντίστοιχη υπηρεσία ονομάζεται Amazon's Elastic Compute Cloud (EC2). Μέσα από μια ιστοσελίδα, ο χρήστης μπορεί να εκκινήσει όσα virtual machines χρειάζεται, με όποιο λειτουργικό σύστημα επιθυμεί και να τα τερματίσει όποτε επιθυμεί. Η χρέωση είναι ανά ώρα και διαφοροποιείται βάση των χαρακτηριστικών του virtual machine, πχ μνήμη, πυρήνες επεξεργαστή. Επίσης, προσφέρει αποθηκευτικό χώρο, ο οποίος ενοικιάζεται με το μήνα καθώς και άλλες υπηρεσίες.

Στο Amazon EC2 ο όρος elasticity έχει εξέχουσα σημασία και προσφέρεται στους χρήστες σε όλα τα επίπεδα. Όπως ήδη αναφέραμε μπορεί να εκκινήσουμε ή να τερματίσουμε virtual machines κατά το δοκούν, να αυξομειώσουμε τον αποθηκευτικό χώρο που νοικιάζουμε κ.α.

Μιας και το Amazon EC2 είναι από τους πρωτοπόρους, με εξαιρετικά ισχυρή παρουσία στον τομέα και υψηλή δημοτικότητα, αλλά και εξαιρετική πληρότητα στις υπηρεσίες που προσφέρει, σε επόμενο κεφάλαιο θα αναφερθούμε εκτεταμένα σε αυτό. [6][7]

Platform as a Service

Στο επίπεδο των υπηρεσιών πλατφόρμας (Platform as a Service, PaaS) παρέχονται υπηρεσίες που ανήκουν στο αμέσως υψηλότερο επίπεδο από το προηγούμενο. Ουσιαστικά πρόκειται για ένα σύνολο δυνατοτήτων, ένα solution stack, που περιλαμβάνει ό,τι χρειάζεται ένας προγραμματιστής για να δημιουργήσει μια εφαρμογή - τόσο από πλευράς συγγραφής κώδικα όσο και από πλευράς εκτέλεσης.

Οι υπηρεσίες πλατφόρμας μπορούν να θεωρηθούν ως η μετεξέλιξη των υπηρεσιών φιλοξενίας εφαρμογών ιστού. Στο πρόσφατο παρελθόν, οι εταιρίες που παρείχαν τέτοιες υπηρεσίες, ουσιαστικά προσέφεραν ένα σύνολο από λύσεις για την ανάπτυξη ιστοσελίδων και κυρίως για την εκτέλεση αυτών. Οι υπηρεσίες πλατφόρμας στο cloud προχωράνε παραπέρα, προσφέροντας ένα πλήρες μοντέλο διαχείρισης του κύκλου ζωής των εφαρμογών, δηλαδή δυνατότητες διαχείρισης όλων των φάσεων ανάπτυξης όπως σχεδιασμό, ανάπτυξη, έλεγχο και εκτέλεση. Έτσι, ελαχιστοποιούν το κόστος διαχείρισης όλων αυτών των περιβαλλόντων που απαιτούν οι ομάδες ανάπτυξης, QA, UAT κλπ.

Από την άλλη πλευρά, η μεταφορά του κύκλου ζωής στο cloud ενέχει περισσότερο από κάθε άλλο επίπεδο και τον κίνδυνο του vendor lock-in. Αυτό συμβαίνει γιατί οι υπηρεσίες στο επίπεδο αυτό είναι πιο συγκεκριμένες σε σχέση με τα άλλα επίπεδα. Όπως είδαμε στο IaaS το βασικό "προϊόν" είναι ένας virtual server - κάτι αρκετά abstract, αφού το λειτουργικό σύστημα αφαιρεί τις ιδιαιτερότητες κάθε μηχανής. Στο επίπεδο της πλατφόρμας όμως, και επειδή οι υπηρεσίες πλατφόρμας δεν είναι ευρέως χρησιμοποιούμενες, ούτε έχουν μεγάλο παρελθόν, το τι προσφέρει ο κάθε πάροχος επαφίεται στον ίδιο, κάτι που φυσικά μεγαλώνει τον κίνδυνο του vendor lock-in.

Ο φόβος του vendor lock-in έχει οδηγήσει στην εμφάνιση ενός νέου είδους PaaS, του Open Platform as a Service. Ουσιαστικά πρόκειται για μια εναλλακτική του PaaS, που προσφέρει ό,τι και αυτό, με τη διαφορά ότι δεν υφίσταται περιορισμούς στην επιλογή λογισμικού και τεχνολογιών ανάπτυξης αποτρέποντας το lock-in.

Μερικοί από τους δημοφιλέστερους παρόχους υπηρεσιών πλατφόρμας είναι το Google App Engine, το Microsoft's Azure, το Heroku.com, το AppJet, το Etelos και το Force.com.[6][7]

Software as a Service

Η πρώτη ίσως υλοποίηση υπηρεσιών τύπου cloud είναι το Software as a Service (SaaS), δηλαδή υπηρεσίες λογισμικού. Πρόκειται για διαδικτυακές εφαρμογές οι οποίες προσφέρουν κάποιες υπηρεσίες. Οι ρίζες του SaaS βρίσκονται στην εποχή όταν διάφορες εταιρίες φιλοξενίας ιστοσελίδων άρχισαν να φιλοξενούν εφαρμογές και η διάδοση του διαδικτύου οδήγησε στην αύξηση της δημοτικότητάς τους. Πρώτος τομέας που επωφεληθήκε ήταν οι εφαρμογές για διαχείριση πωλήσεων και πελατειακών σχέσεων, CRM. Εταιρίες που ειδικεύονταν σε πωλήσεις χωρίς να έχουν άμεση σχέση με την πληροφορική βρήκαν απόλυτα χρηστική τη δυνατότητα να έχουν τα οφέλη της μηχανογράφησης αφαιρώντας το πρόβλημα της διαχείρισης των

υπολογιστικών συστημάτων. Πάροχοι υπηρεσιών λογισμικού, όπως η [salesforce.com](https://www.salesforce.com) έχτισε την πορεία της πουλώντας CRM λύσεις σαν SaaS.

Το μοντέλο σε αυτό το επίπεδο είναι το εξής. Μια εταιρία παρέχει κάποιες εφαρμογές μέσω του διαδικτύου. Μια άλλη εταιρία που χρειάζεται να χρησιμοποιήσει κάποιου είδους λογισμικό, αντί να αγοράσει το λογισμικό, servers και να διατηρεί IT τμήμα για τη συντήρησή του, ενοικιάζει το δικαίωμα χρήσης από την πρώτη εταιρία η οποία το χρεώνει με βάση τη χρήση και τον αριθμό των χρηστών. Έτσι, η πρώτη εταιρία επωφελείται κάνοντας οικονομίες κλίμακας και κερδίζοντας από τους χρήστες, ενώ η δεύτερη μειώνει το λειτουργικό της κόστος και αυξάνει δραστικά την απόδοση της επένδυσής της (ROI).[6][7]

Hardware as a Service

Τελευταίο αφήσαμε το hardware as a service, εξαιτίας του ότι δεν χρησιμοποιείται ιδιαίτερος. Μάλιστα, πολλές πηγές δεν το αναφέρουν καν. Το αναφέρουμε απλώς για λόγους πληρότητας.

Στο μοντέλο αυτό, ο πάροχος επιτρέπει στους πελάτες του να ενοικιάζουν hardware. Με άλλα λόγια, πρόκειται για την ενοικίαση υλικού, που επιτρέπει στους χρήστες τη δημιουργία datacenters χωρίς να πρέπει να αγοράσουν το υλικό που απαιτείται.[6]

Ταξινόμηση συστημάτων Cloud βάση τοπολογίας

Αφού μελετήσαμε πώς ταξινομούνται οι υπηρεσίες cloud και κατ' επέκταση τα συστήματα Cloud βάσει των υπηρεσιών που προσφέρουν, θα δούμε πώς ταξινομούνται τα συστήματα αυτά βάση της τοπολογίας τους και της προσβασιμότητάς τους.

Κάποιος θα περίμενε, πως βάση της δομής τους τα συστήματα cloud είναι δημόσια, με τον καθένα να μπορεί να τα χρησιμοποιεί. Αν και αυτό αληθεύει, η αξία των δεδομένων που διαχειρίζεται μια εταιρία, συχνά απαγορεύει την τοποθέτησή τους σε δημόσιους χώρους. Έτσι, δεν αποτελεί έκπληξη το γεγονός ότι εκτός από τα δημόσια, υπάρχουν και ιδιωτικά clouds, όπως και υβριδικά, δηλαδή συστήματα cloud που συνδυάζουν ένα ή περισσότερα ιδιωτικά με ένα ή περισσότερα δημόσια. Επιπλέον, η πρόσφατη εμφάνιση εικονικών ιδιωτικών clouds (Virtual Private Clouds, VPCs) τα οποία ζουν σε δημόσια κάνει το τοπίο πιο περίπλοκο.

Για το λόγο αυτό, θα εισαγάγουμε τον όρο εσωτερικά (internal) συστήματα cloud ο οποίος είναι ισοδύναμος των ιδιωτικών clouds και αφορά τα συστήματα τα οποία βρίσκονται στο εσωτερικό της εταιρίας.

Πρέπει να τονιστεί πως η ταξινόμηση αυτή είναι ανεξάρτητη από το επίπεδο υπηρεσιών.

Δημόσια συστήματα cloud

Πρόκειται για clouds τα οποία είναι προσβάσιμα σε όλους και παρέχουν υπηρεσίες σε οποιονδήποτε το ζητήσει. Αυτό φυσικά δεν καταρρίπτει την ιδιωτικότητα, αφού κάθε χρήστης-πελάτης έχει το δικό “χώρο” στον οποίο δεν έχουν πρόσβαση οι υπόλοιποι χρήστες-πελάτες. Οι πάροχοι που προαναφέραμε προσφέρουν όλοι δημόσια cloud συστήματα.

Εάν δεν υπάρχουν περιορισμοί, όπως θα δούμε παρακάτω, μια εταιρία μπορεί να κατευθυνθεί προς αυτή τη λύση ώστε να επωφεληθεί:

- Όταν, οι εφαρμογές της εταιρίας χρησιμοποιούνται από πολλούς ανθρώπους σε διαφορετικά γεωγραφικά σημεία. Κλασικότερο παράδειγμα το email.
- Για την ανάπτυξη και το testing των εφαρμογών της
- Έχει εφαρμογές με μεγάλες διακυμάνσεις στη χρήση τους και το φόρτο

Ιδιωτικά (εσωτερικά) συστήματα cloud

Παρότι η ιδέα της χρήσης ενός δημοσίου cloud είναι αρκετά ελκυστική, εξαιτίας της μείωσης του κόστους συντήρησης και λειτουργίας τους, υπάρχουν εταιρίες που κινούνται προς τη χρήση εσωτερικών συστημάτων cloud. Ανάμεσα στους λόγους που θα αναζητήσουμε βρίσκεται:

- η ήδη μεγάλη επένδυση σε υπολογιστικά συστήματα
- η εξαιρετική σημασία της ασφάλειας των δεδομένων, πχ οικονομικά
- νομικοί περιορισμοί
- το μέγεθος της εταιρίας που της επιτρέπει να προβεί σε οικονομίες κλίμακος ώστε να είναι συμφέρον να διατηρεί ένα εσωτερικό σύστημα cloud

Τα συστήματα αυτά τρέχουν στο εσωτερικό της εταιρίας, σε ιδιόκτητα μηχανήματα και η διαχείρισή τους γίνεται από προσωπικό της εταιρίας.

Εικονικά ιδιωτικά συστήματα cloud

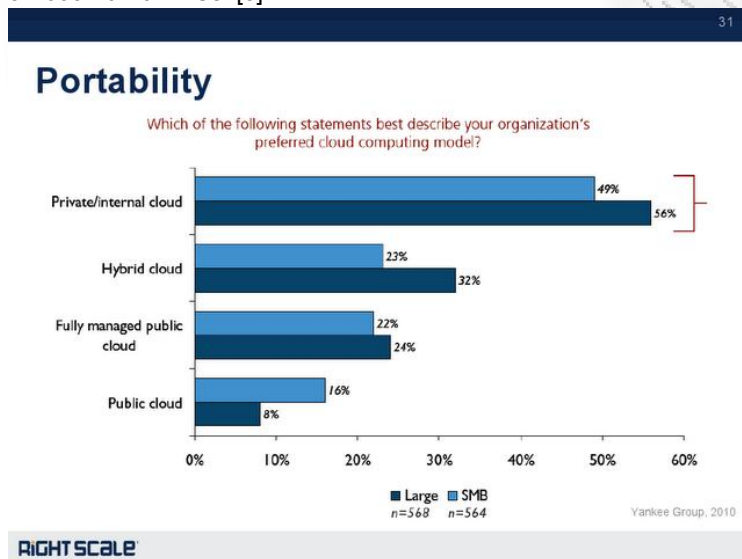
Αντιλαμβανόμενοι τα ζητήματα ασφαλείας των δημοσίων clouds, οι μεγαλύτεροι πάροχοι, όπως η amazon και το [salesforce.com](https://www.salesforce.com) έσπευσαν να προσφέρουν μια ενδιαμέση λύση, τα εικονικά ιδιωτικά clouds, VPCs. Αν το όνομα παραπέμπει στα VPNs, αυτό όντως έχει σχέση, αφού οι λύσεις και των δύο εταιριών αξιοποιούν την εν λόγω τεχνολογία. Αυτό που προσφέρουν είναι απομονωμένους υπολογιστικούς πόρους - στους οποίους δεν έχουν τρίτοι πρόσβαση- οι οποίοι διασυνδέονται μέσω VPN ώστε οι χρήστες-πελάτες να τους “βλέπουν” ως μέρος του ιδιωτικού τους δικτύου.

Υβριδικά συστήματα cloud

Το τελευταίο είδος είναι τα υβριδικά συστήματα, δηλαδή συνδυασμός public clouds με private clouds (είτε εσωτερικά είτε εικονικά). Πρόκειται για μια λύση που γίνεται αρκετά δημοφιλής αφού θεωρητικά συνδυάζει τα μέγιστα οφέλη και από τις δύο πλευρές. Έτσι, οι εφαρμογές για τις εσωτερικές επιχειρησιακές διαδικασίες φιλοξενούνται στο private cloud, μεγιστοποιώντας την ταχύτητα του δικτύου, αποφεύγοντας νομικά και άλλα ζητήματα όσον αφορά τα δεδομένα και διαφυλάσσοντας ταυτόχρονα τις επενδύσεις στον υπάρχοντα εξοπλισμό, ενώ οι εξωτερικές εφαρμογές που προσφέρονται σε χρήστες φιλοξενούνται στο δημόσιο cloud, αφαιρώντας φόρτο εργασίας από τα εσωτερικά συστήματα και μειώνοντας το κόστος συντήρησης και διαχείρισης.

Στα πλαίσια των υβριδικών συστημάτων έχουν προταθεί και λύσεις ακόμα πιο σύνθετες, όπως η εκτέλεση των δικτυακών εφαρμογών σε δημόσια clouds με τη διατήρηση κάποιων συστημάτων, όπως οι βάσεις δεδομένων στο ιδιωτικό cloud.

Στο επόμενο διάγραμμα φαίνονται οι τάσεις της αγοράς. Ως fully managed public cloud εννοούνται τα VPCs. [8]



Το οικονομικό μοντέλο του cloud computing

Μιας και το cloud computing είναι περισσότερο θέμα management, αξίζει να εξετάσουμε συνοπτικά πώς λειτουργεί από οικονομικής άποψης και γιατί είναι συμφέρον τόσο για τους παρόχους όσο και για τους χρήστες. Ο όρος κλειδί εδώ είναι οι “οικονομίες κλίμακας”. Με τον όρο αυτόν νοείται η μείωση του κόστους που επιτυγχάνει μια επιχείρηση αυξάνοντας την ποσότητα του παραγόμενου προϊόντος και χαρακτηρίζει την παραγωγή ενός οποιουδήποτε

προϊόντος. Τα οφέλη από τις “οικονομίες κλίμακας” είναι η μείωση του διοικητικού κόστους, οικονομίες έρευνας, αύξηση διαπραγματευτικής δύναμης κ.α. [9]

Έτσι, μια εταιρία που ασχολείται αποκλειστικά με τη διαχείριση ενός συστήματος cloud στο οποίο φιλοξενούνται δεκάδες άλλα υπολογιστικά συστήματα έχει τη δυνατότητα να μειώσει δραστικά το κόστος διαχείρισης και συντήρησης ανά υπολογιστικό σύστημα.

Σύμφωνα με έρευνες, σε ένα παραδοσιακό datacenter είναι εφικτή η διαχείριση έως 150 servers ανά administrator. Νούμερο που πέφτει δραματικά σε μικρές εταιρίες. Αντίθετα, σε clouds το νούμερο ανεβαίνει σε μερικές χιλιάδες. [10]

15

Automation

Operational efficiency

- Server-to-admin ratio is an indicator of admin costs
 - Inefficient operations as low as 20:1
 - Above average ratio 150:1 (enterprises typically in the 70 to 140 range)
 - Best practices over 2,000:1
- Savings on admin costs of easily 50%

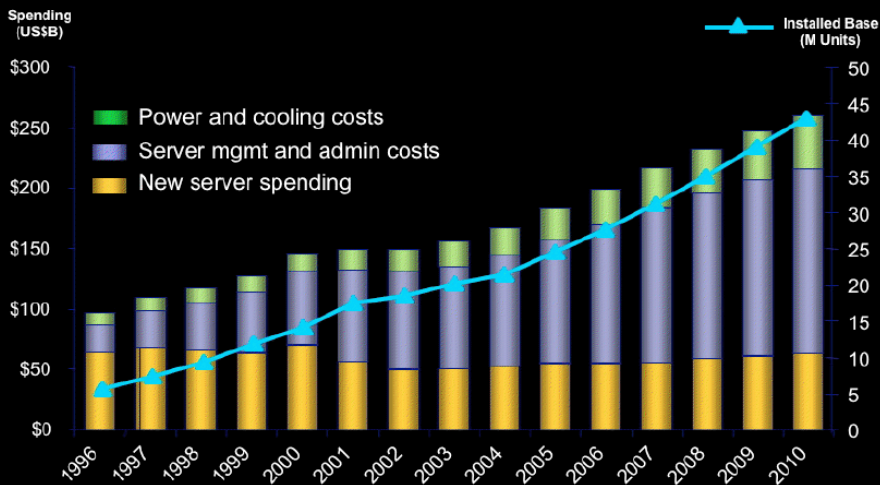
Source: Deepak Patil, GFS 2006

RIGHTSCALE

Αν σε αυτό συνυπολογίσουμε το μειωμένο κόστος κτήσης που προφανώς έχουν οι εταιρίες αυτές, μιας και αγοράζουν υπολογιστικά συστήματα αλλά και λογισμικό με σαφώς μεγαλύτερους ρυθμούς και σε μεγαλύτερες ποσότητες, το μειωμένο κόστος λειτουργίας σε ρεύμα, ψύξη κλπ αλλά και το γεγονός ότι έχουν εφαρμογές ειδικά σχεδιασμένες για μαζική διαχείριση, γίνεται ξεκάθαρο πώς μπορούν να έχουν εξαιρετικά μειωμένο κόστος ανά υπολογιστικό σύστημα.

Όπως φαίνεται από το παρακάτω διάγραμμα, για ένα παραδοσιακό IT, αυτό που αυξάνει συνεχώς είναι κόστος διαχείρισης και λειτουργίας - αυτό ακριβώς το οποίο έχουν μειώσει οι πάροχοι clouds. Έτσι, κάποιο από το όφελος αυτό περνά στον πελάτη, ώστε και οι δύο πλευρές είναι ωφελημένες.

IT infrastructure growth



IDC - Virtualization 2.0 - Dec 2006

RIGHT SCALE

Συμπεράσματα

Το "cloud computing" λοιπόν είναι ένα σύνολο τεχνολογικών προσεγγίσεων στη διαχείριση των υπολογιστικών πόρων. Εταιρίες παροχής υπηρεσιών cloud computing μέσω των μεγθών των υπολογιστικών τους κέντρων καταφέρνουν να επιτύχουν αξιοσημείωτες οικονομίες κλίμακας μειώνοντας δραστικά το κόστος διαχείρισης και συντήρησης ανά υπολογιστικό σύστημα. Μέρος της μείωσης αυτής μεταφέρεται στους υποψηφίους πελάτες-χρήστες οι οποίοι με τη σειρά τους μειώνουν το κόστος διαχείρισης προσφέροντας ταυτόχρονα στις εταιρίες που εργάζονται και στους δικούς τους πελάτες μεγαλύτερη ευελιξία.

Επειδή, όμως η διαχείριση είναι παρούσα σε όλες τις φάσεις του κύκλου ζωής του λογισμικού και δεν αφορά μόνο στις εφαρμογές που "τρέχουν live", θα εξετάσουμε το πώς μπορεί το cloud computing να ωφελήσει τον τομέα ανάπτυξης λογισμικού βελτιώνοντας τη διαχείριση των περιβαλλόντων ανάπτυξης, QA και UAT.

Το οικοσύστημα ανάπτυξης λογισμικού

Εισαγωγή

Πολλά έχουν αλλάξει τα τελευταία χρόνια στον τομέα της ανάπτυξης λογισμικού. Έχοντας εγκαταλείψει προ πολλού τον παραδοσιακό τρόπο ανάπτυξης, με τον προγραμματιστή να γράφει κώδικα, να τον μεταγλωττίζει και να τον εκτελεί τοπικά, ελέγχοντας τη λειτουργία της εφαρμογής και κάνοντας debugging, η ανάπτυξη λογισμικού έχει μετακινηθεί προς περισσότερο ευέλικτες μεθοδολογίες. Επιπλέον, οι απαιτήσεις ασφαλείας έχουν αλλάξει την νοοτροπία “ο,τι παίζει δεν το πειράζουμε” σε “ό,τι παίζει, για να συνεχίσει να παίζει, πρέπει να αναβαθμίζεται”.

Ταυτόχρονα, οι εφαρμογές παίζουν όλο και μεγαλύτερο και σημαντικότερο ρόλο στη λειτουργία των επιχειρήσεων, αποτελώντας αναντικατάστατα πλέον εργαλεία σε όλα τα επίπεδα επιχειρησιακών διαδικασιών. Αυτό όμως επέφερε και ισχυρότερες απαιτήσεις για την ευελιξία του λογισμικού, ώστε να μην αποτελεί τροχοπέδη στην εξέλιξη της εταιρίας.

Προκειμένου να υποστηριχθεί όλη αυτή η απαίτηση για λογισμικό που εξελίσσεται συνεχώς, έχουν αναπτυχθεί νέα συστήματα που απαλλάσσουν τους προγραμματιστές από δευτερεύουσες εργασίες όπως η επίλυση εξαρτήσεων, η αναβάθμιση βιβλιοθηκών, η συντήρηση πολλαπλών εκδόσεων και η αντιμετώπιση προβλημάτων λόγω αναβαθμίσεων της εφαρμογής.

Αν στα παραπάνω προσθέσουμε και τη ραγδαία ανάπτυξη του ανοιχτού λογισμικού που γίνεται κατ' εξοχήν κατανεμημένα, από ομάδες σε όλο τον κόσμο, και επέφερε πρόσθετες απαιτήσεις στην επικοινωνία και τη συνεργασία των μελών των ομάδων αντιλαμβανόμαστε πως το οικοσύστημα ανάπτυξης λογισμικού έχει μεταλλαχθεί δραματικά.

Πλέον, οι νέες μεθοδολογίες προτείνουν πιο ευέλικτα μοντέλα ανάπτυξης και απαιτούν ένα οικοσύστημα που μπορεί να τα υποστηρίξει. Έτσι, έχουν αναπτυχθεί εργαλεία διαχείρισης των εξαρτήσεων, οι δοκιμές τείνουν να αυτοματοποιούνται συνδυάζοντας τη χρήση κατάλληλων εργαλείων και τον κώδικα που γράφεται αποκλειστικά για το λόγο αυτό και αποτελούν τμήμα του build, τα παραδοσιακά εργαλεία ελέγχου εκδόσεων κώδικα (version control systems) και αυτά της παρακολούθησης των σφαλμάτων (bug tracking systems) τείνουν να συνεργάζονται και να παρέχουν μηχανισμούς ενσωμάτωσής τους στα περιβάλλοντα ανάπτυξης (Integrated Development Environments, IDE). Επιπλέον, έχουν αναπτυχθεί εργαλεία ελέγχου της ποιότητας του κώδικα καθώς και της απόδοσης αυτού.

Εργαλεία ανάπτυξης

Τα ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDE) αποτελούν ίσως το πιο σταθερό κομμάτι στον ταχύτατα μεταβαλλόμενο κόσμο της ανάπτυξης λογισμικού. Φυσικά, υφίστανται βελτιώσεις, αποκτούν περισσότερες λειτουργίες, έχουν μετατραπεί από απλοί κειμενογράφοι σε προχωρημένα συστήματα που βοηθούν αισθητά το χρήστη στη συγγραφή του κώδικα, ενσωματώνουν λειτουργίες για συνεργασία με εξωτερικά συστήματα και πλέον οδεύουν κι αυτά προς το διαδίκτυο. Όπως θα δούμε, υπάρχουν αρκετές προτάσεις για τη συγγραφή κώδικα στο διαδίκτυο, ακόμα και για μη διερμηνευόμενες γλώσσες.

Βέβαια, απαιτείται χρόνος για να αναπτυχθούν εκ νέου αντίστοιχα συστήματα με αυτά που υπάρχουν, λόγω κυρίως της αυξημένης λειτουργικότητας που αυτά ενσωματώνουν. Από την άλλη, αρκετά από αυτά που υπάρχουν έχουν τη δυνατότητα να μετατραπούν σε εφαρμογές ιστού, πχ java desktop εφαρμογές σε java applets. Εν τούτοις, η τάση είναι να εκτελούνται όλα από ιστοσελίδες, χωρίς να απαιτείται η εγκατάσταση άλλων μηχανών πχ jvm στον client.

Build systems

Η αύξηση του μεγέθους των εφαρμογών, επέφερε προβλήματα και στη διαδικασία μεταγλώττισης. Πλέον, το παραδοσιακό compile δεν αρκεί, αλλά αποτελεί μέρος μιας ευρύτερης διαδικασίας με το όνομα build. Στη διαδικασία αυτή συμπεριλαμβάνεται φυσικά η μεταγλώττιση (compile) αλλά και η εκτέλεση αυτοματοποιημένων δοκιμών.

Μία από τις προτάσεις, για την αντιμετώπιση των συνεχών αλλαγών στις απαιτήσεις, η οποία βρήκε γρήγορα αποδοχή ήταν το continuous integration. Πρόκειται για μια προσέγγιση κατά την οποία πραγματοποιούνται επαναλαμβανόμενα builds σε κάθε αλλαγή του κώδικα. Μαζί με τα builds βέβαια εκτελούνται και τα αυτοματοποιημένα σενάρια ελέγχου. Έτσι, ακολουθώντας την αρχή του “fail early” είναι δυνατή η πρόωμη αναγνώριση λαθών και προβλημάτων και η γρήγορη αντιμετώπισή τους.

Πλέον, το αυτοματοποιημένο build εκτελείται από τους CI (Continuous Integration) servers. Επιπλέον, αποθηκεύονται τα εξαγόμενα του build ώστε να είναι διαθέσιμα ανά πάσα στιγμή και στις υπόλοιπες ομάδες, όπως QA ή UAT. Με αυτό τον τρόπο βελτιώνεται δραστικά το κενό ανάμεσα στη συγγραφή του κώδικα και τον ποιοτικό του έλεγχο, με τις ομάδες QA και UAT να παρακολουθούν στενά την εξέλιξη του έργου και να διαπιστώνουν νωρίς πιθανά προβλήματα.

Dependency management

Ένα άλλο πρόβλημα που επήλθε με την αύξηση του μεγέθους των εφαρμογών αλλά και την απαίτηση για ενημερωμένο και ασφαλές λογισμικό ήταν η συντήρηση των εξαρτήσεων από βιβλιοθήκες.

Πλέον, υπάρχει διαθέσιμο άφθονο λογισμικό, με τη μορφή βιβλιοθηκών, το οποίο χρησιμοποιείται ευρύτατα, ώστε η κάθε ομάδα ανάπτυξης να εστιάζει στα επιχειρησιακά ζητήματα. Έτσι, οι βιβλιοθήκες αφ' ενός παρέχουν έτοιμη λειτουργικότητα, αφ' εταίρου η συντήρησή τους γίνεται από αυτούς που τις κατασκευάζουν και όχι από αυτούς που τις χρησιμοποιούν. Επιπρόσθετα, η αυξημένη βάση χρηστών μιας δημόσιας - και ακόμη περισσότερο - ανοιχτού κώδικα βιβλιοθήκης, εξασφαλίζει την ορθότητα της λειτουργίας της, τη συνεχή διόρθωση και βελτίωσή της και τον διαρκή εμπλουτισμό της με νέα χαρακτηριστικά. Για να έχουν νόημα αυτά βέβαια, απαιτείται όσοι χρησιμοποιούν αυτές τις βιβλιοθήκες να χρησιμοποιούν τις τελευταίες εκδόσεις.

Από την άλλη μεριά, και οι ίδιες οι βιβλιοθήκες χρησιμοποιούν άλλες βιβλιοθήκες. Χαρακτηριστικότερο ίσως παράδειγμα αποτελούν οι βιβλιοθήκες καταγραφής (logging), κοινή απαίτηση για όλες σχεδόν τις εφαρμογές. Έτσι, οι χρήστες των βιβλιοθηκών θα πρέπει να λαμβάνουν υπόψιν και τις εξαρτήσεις των βιβλιοθηκών, οι οποίες ενδέχεται να μεταβάλλονται από έκδοση σε έκδοση.

Λαμβάνοντας υπόψιν όλα τα παραπάνω, γίνεται αντιληπτό πως η επίλυση και η συντήρηση των εξαρτήσεων είναι ένα σοβαρό ζήτημα που, αν γίνεται με μη αυτόματο τρόπο, απαιτεί πολλές εργατοώρες και φυσικά είναι επιρρεπής σε λάθη. Για το λόγο αυτό έχουν αναπτυχθεί συστήματα στα οποία δηλώνονται οι εξαρτήσεις για κάθε κομμάτι του λογισμικού, καθώς και η φάση που απαιτούνται, πχ compile, testing, execution. Τα συστήματα αυτά, μέρος των build συστημάτων, με ευρύτερα χρησιμοποιούμενα το maven και το ant, αναλαμβάνουν να “κατεβάσουν” τις σωστές εκδόσεις των βιβλιοθηκών ώστε να είναι διαθέσιμες εκεί που απαιτούνται.

Testing: QA & UAT

Μία από τις απαιτήσεις των ευέλικτων (agile) μεθοδολογιών είναι ότι το σύστημα θα πρέπει ανά πάσα στιγμή να είναι έτοιμο να “βγει στην παραγωγή” (deployment) με τα features φυσικά που έχουν αναπτυχθεί μέχρι εκείνο το σημείο. Αυτό συνεπάγεται ότι το testings, τόσο από πλευράς QA όσο και από πλευράς UAT θα πρέπει να είναι συνεχές ούτως ώστε να είναι πάντα εξασφαλισμένη η αξιοπιστία του λογισμικού. Αυτό όμως προϋποθέτει πως τα περιβάλλοντα QA και UAT ενημερώνονται διαρκώς με τις νέες αλλαγές, κάτι που έχει αυξήσει αισθητά το κόστος διαχείρισής τους. Επιπλέον, συχνά είναι απαραίτητος ο έλεγχος σφαλμάτων σε παλαιότερες εκδόσεις που ενδέχεται να βρίσκονται ακόμα σε λειτουργία. Αυτό σημαίνει πως θα πρέπει να είναι εύκολη και άμεση η χρήση ενός περιβάλλοντος με την αντίστοιχη έκδοση.

Γίνεται προφανές λοιπόν, πως οι απαιτήσεις για ευελιξία έχουν επηρεάσει σημαντικά και τις φάσεις QA και UAT, αυξάνοντας το κόστος διαχείρισης πολλών διαφορετικών περιβαλλόντων, με διαφορετικές εκδόσεις της εφαρμογής. Στο σημείο αυτό, η χρήση των υπηρεσιών cloud δείχνει να ταιριάζει απόλυτα, αφού παρέχει τη δυνατότητα της αποθήκευσης instances με συγκεκριμένες εκδόσεις που θα εκτελούνται κατ' απαίτηση. Με τον τρόπο αυτό επιταχύνονται οι διαδικασίες ελέγχου και μειώνεται το κόστος λειτουργίας.

Συμπεράσματα

Η εξέλιξη του λογισμικού σε ένα αναπόσπαστο κομμάτι των επιχειρησιακών λειτουργιών έχει επιφέρει αυξημένες απαιτήσεις στην ανάπτυξη λογισμικού. Η ανάπτυξη οφείλει να είναι ταχύτερη και πιο ευέλικτη. Επιπλέον, το λογισμικό έχει γίνει πολυπλοκότερο και οι εφαρμογές μεγαλύτερες, ενώ η συνεχής προσφορά έτοιμων βιβλιοθηκών έχει αυξήσει τις εξαρτήσεις σε εξωτερικό λογισμικό. Έτσι, έχουν αναπτυχθεί μεθοδολογίες που προτείνουν ευέλικτους τρόπους ανάπτυξης αλλά και συστήματα για την υποστήριξή τους. Ταυτόχρονα, ο έλεγχος ποιότητας του λογισμικού γίνεται εντονότερος και αποκτά όλο και μεγαλύτερη σημασία. Όλα αυτά οδηγούν στην αύξηση του κόστους και του χρόνου διαχείρισης των υπολογιστικών συστημάτων που απαιτούνται για την ανάπτυξη και τον έλεγχο.

Το cloud computing θα μπορούσε να παίξει κι εδώ σημαντικό ρόλο, μειώνοντας το κόστος και το χρόνο διαχείρισης των συστημάτων αυτών, επιταχύνοντας τις διαδικασίες και τη μετάβαση από τη μια φάση ανάπτυξης στην επόμενη. Από την άλλη μεριά, βέβαια, θέτει κι αυτό τα δικά του ζητήματα αφού οι εφαρμογές θα πρέπει να σχεδιάζονται λαμβάνοντας υπόψιν και τις δυνατότητες κλιμάκωσης που προσφέρονται μέσω του cloud. Παρακάτω θα παρουσιάσουμε τις πλατφόρμες ανάπτυξης λογισμικού στο cloud συμπεριλαμβανομένων τόσο των συστημάτων συγγραφής κώδικα όσο και πλατφορμών εκτέλεσης που μπορούν να χρησιμοποιηθούν για QA και UAT - δύο αναπόσπαστες φάσεις του κύκλου ανάπτυξης.

Πλατφόρμες συγγραφής κώδικα στο cloud

Εισαγωγή

Μία από τις προωθούμενες αλλαγές που φέρνει το cloud στην ανάπτυξη εφαρμογών αφορά τη συγγραφή και μεταγλώττιση του κώδικα. Ενώ υπήρχαν ήδη βιβλιοθήκες javascript για τη μορφοποίηση του κώδικα που παρουσιάζεται σε ιστοσελίδες, καθώς και κάποια εργαλεία συγγραφής - κυρίως javascript - κώδικα, έχει αρχίσει η ανάπτυξη ολοκληρωμένων περιβαλλόντων ανάπτυξης (Integrated Development Environments, IDE) σε javascript.

Αυτά τα εργαλεία προσομοιάζουν σε μια ιστοσελίδα τις λειτουργίες ενός κλασσικού IDE, παρέχοντας - ή καλύτερα, στοχεύοντας στο να παρέχουν - στον προγραμματιστή όλα τα εργαλεία και τις ευκολίες που έχει συνηθίσει να βρίσκει σε ένα μοντέρνο περιβάλλον ανάπτυξης, όπως αυτόματη συμπλήρωση κώδικα, βοηθός συγγραφής και πρότυπα κώδικα, αυτοματοποίηση ελέγχων και builds, εντοποίηση με συστήματα διαχείρισης εκδόσεων κώδικα (Version Control Systems, όπως subversion, git κ.α.), documentation κλπ.

Οι προφανείς γλώσσες στις οποίες στοχεύουν αυτά τα εργαλεία είναι οι διερμηνευόμενες (Interpreted), όπως PHP, Ruby, Javascript. Το πλεονέκτημα ανάπτυξης σε αυτές τις γλώσσες είναι πως απαλείφουν την ανάγκη μεταγλώττισης του κώδικα. Αντίθετα, η απλή αποθήκευση του αρχείου με τον πηγαίο κώδικα είναι αρκετή ώστε να είναι δυνατή η εκτέλεση του κώδικα. Εν τούτοις, στις προσπάθειες που θα παρουσιάσουμε υπάρχουν και αυτές που υποστηρίζουν μη διερμηνευόμενες γλώσσες, όπως η Java. Πρακτικά, αυτό που συμβαίνει είναι ότι το compilation - ή build - λαμβάνει χώρα στο server. Ταυτόχρονα, γίνεται και deploy της εφαρμογής ώστε να είναι διαθέσιμη σε κάποιο url - όπως συμβαίνει και στα κλασσικά περιβάλλοντα εργασίας.

Τα εργαλεία που παρουσιάσουμε είναι το codility, το kodingen, το cloud-ide με το cloudbees και το salesforce.com. Το codility είναι μια εφαρμογή για την αξιολόγηση προγραμματιστών. Στον προγραμματιστή ανατίθεται ένα πρόβλημα, το οποίο μπορεί να επιλύσει χρησιμοποιώντας τη γλώσσα της επιλογής του. Το σύστημα μεταγλωττίζει και εκτελεί τον κώδικα, παρουσιάζοντας τα αποτελέσματα στον προγραμματιστή. Το kodingen είναι μια πλατφόρμα ανάπτυξης ιστοσελίδων σε php και ruby. Μέσα από το περιβάλλον που παρέχει είναι δυνατή, εκτός φυσικά από την ανάπτυξη μιας εφαρμογής και η εγκατάσταση έτοιμων εφαρμογών και βιβλιοθηκών. Το cloud-ide παρέχει τη δυνατότητα για τη ανάπτυξη εφαρμογών Java παρέχοντας σημαντικά εργαλεία, όπως continuous integration servers και μηχανισμούς ελέγχου ποιότητας κώδικα, ενώ συνεργάζεται με την πλατφόρμα του cloudbees. Τέλος, το salesforce.com είναι η προσπάθεια του ομώνυμου site να εισέλθει στο χώρο του cloud.

Παρουσίαση

Codility

Το codility είναι ένα εργαλείο για αυτοματοποιημένη αξιολόγηση προγραμματιστικών ικανοτήτων. Έχει σχεδιαστεί για εταιρίες ανάπτυξης λογισμικού αλλά και για εταιρίες με υψηλές απαιτήσεις σε προσωπικό τεχνολογίας υπολογιστών (IT staff). Επιτρέπει σε recruiters, ακόμα κι αν δεν έχουν ιδιαίτερες γνώσεις στην ανάπτυξη λογισμικού, να αξιολογούν υποψηφίους, επιβεβαιώνοντας τις προγραμματιστικές ικανότητες αυτών με έναν αυτοματοποιημένο τρόπο. Ο στόχος του codility είναι να επιβεβαιώνει το πόσο καλά μπορεί να αποδίδει ένας προγραμματιστής στα καθημερινά του καθήκοντα.

Το codility αξιολογεί τον υποψήφιο σε πραγματικές συνθήκες, ζητώντας του υλοποιήσει μια ρουτίνα για την επίλυση ενός προβλήματος. Ο προγραμματιστής μπορεί να επιλέξει μια γλώσσα από τις διαθέσιμες και να γράψει τον κώδικα. Έπειτα, το codility ελέγχει την ορθότητα καθώς και τις επιδόσεις του κώδικα. Ουσιαστικά, προσομοιώνει ένα περιβάλλον ανάπτυξης επιτρέποντας στον υποψήφιο να μεταγλωττίζει και να τρέχει τον κώδικα, παρέχοντας και δεδομένα δοκιμών ώστε να μπορεί και ο ίδιος να ελέγχει τον κώδικά του πριν τον αποστείλει στην εταιρία για την οποία δοκιμάζεται.

Σύμφωνα με το ίδιο το codility, “Codility saves time of software talent recruiters by filtering out job candidates who cannot write correct programs. Codility saves time of software talent recruiters by filtering out job candidates who cannot write correct programs. Our customers filter out up to 90% of candidates. Less wasted time. Better hires.”

Ο τρόπος λειτουργίας του codility είναι ο εξής: η εταιρία μπορεί να δημιουργήσει ή να επιλέξει ένα τεστ για κάποιον υποψήφιο και να του στείλει τη διεύθυνση. Εναλλακτικά, δύναται κάποιος προγραμματιστής να αυτοαξιολογηθεί εκ των προτέρων.

Ξεκινώντας τη διαδικασία βλέπουμε την παρακάτω οθόνη η οποία εξηγεί τη διαδικασία.

The screenshot shows the Codility website interface for a programming task. The browser address bar shows the URL: `codility.com/test/c/certJMQWYD-HF8B3MT2KQ5CD4R8/`. The page title is "Codility" and the version is "1.1.20". The time displayed is "00:59:33".

The interface is divided into several sections:

- Task Description:** A non-empty zero-indexed array A consisting of N integers is given. Two integers P and Q are called adjacent in array A if there exists an index $0 \leq K < N - 1$ such that:
 - $P = A[K]$ and $Q = A[K+1]$, or
 - $Q = A[K]$ and $P = A[K+1]$.
 A non-empty zero-indexed sequence B consisting of M integers is called adjacent in array A if the following conditions hold:
 - $B[0] = A[0]$,
 - $B[M-1] = A[M-1]$,
 - $B[K]$ and $B[K+1]$ are adjacent in array A for $0 \leq K < M - 1$.
 For example, consider array A consisting of eight elements such that:

$A[0] = 1$	$A[1] = 10$	$A[2] = 6$
$A[3] = 5$	$A[4] = 10$	$A[5] = 7$
$A[6] = 5$	$A[7] = 2$	

 The following sequences are adjacent in array A :

$[1, 10, 6, 5, 10, 7, 5, 2]$
$[1, 10, 7, 5, 2]$
$[1, 10, 6, 5, 10, 6, 5, 10, 7, 5, 2]$
$[1, 10, 5, 2]$

 The last sequence is the shortest possible sequence adjacent in array A .
- Code Editor:** A Java code editor with the following code:


```

1 // you can also use imports, for example:
2 // import java.math.*;
3 class Solution {
4     public int shortest_adj_seq ( int[] A ) {
5         // write your code here
6     }
7 }
8 
```
- Buttons:** "help", "verify", "submit task", "quit".
- Callouts:**
 - 1: Read the problem description.
 - 2: Choose the programming language.
 - 3: Write the solution for the problem using the editor, your solution should include a function defined in the task description, your solution can include other functions, procedures or methods.
 - 4: Verify correctness of your solution (you will be informed whether your solution (a) compiles, (b) returns a correct result given the exemplary data from the task description, (c) terminates within an acceptable period of time). Verification will help you to avoid simple mistakes. You can verify your solution multiple times. The number of verification requests will not influence your score.
 - 5: You can run your solution on your test data to see what result it computes, you will not be informed about the correctness of this result.
 - 6: Submit the solution for the final evaluation. This will end this session. You can submit your solution only once.
 - 7: Quit will finish this session without sending your solution for evaluation. You can re-open the session, but quitting does not put the timer on hold.

Σχόλιο [1]: hmichopoulos:
First page in codility.com

Εργαζόμενοι στο codility, μπορούμε να υλοποιήσουμε τη ρουτίνα που επιλύει το πρόβλημα που περιγράφεται αριστερά, στη γλώσσα της επιλογής μας (πάνω δεξιά). Επίσης, μπορούμε να ορίσουμε και δεδομένα δοκιμών κάτω αριστερά. Πατώντας verify, το σύστημα θα μεταγλωττίσει και θα εκτελέσει τον κώδικα, παρουσιάζοντας μας τα αποτελέσματα για κάθε δεδομένο εισόδου.

The screenshot shows the Codility website interface. On the left, there is a problem description for 'Shortest Adjacent Sequence'. The description asks for a function that returns the length of the shortest sequence of adjacent integers in an array. It provides an example array [1, 10, 6, 5, 10, 7, 5, 2] and states that the function should return 4. The code editor on the right contains the following Java code:

```
1 // you can also use imports, for example:
2 // import java.math.*;
3 class Solution {
4     public int shortest_adj_seq ( int[] A ) {
5         return 4;
6     }
7 }
8
```

At the bottom of the editor, there is a status bar indicating 'Compilation successful'.

Kodingen

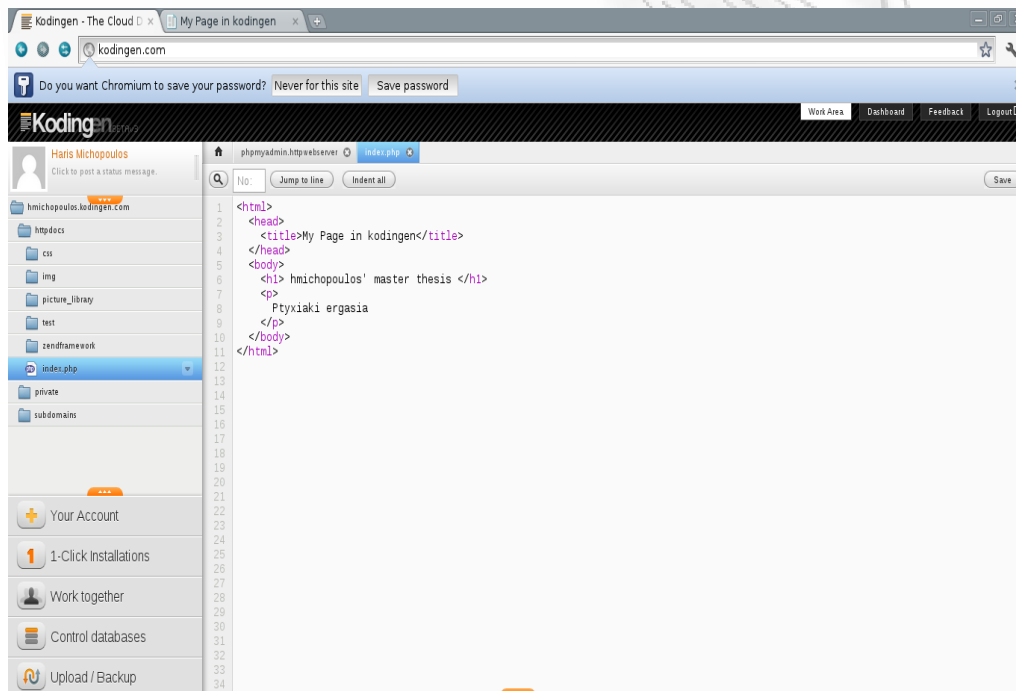
Το kodingen ξεκίνησε ως μια προσπάθεια δυο αδερφών. Το κίνητρό τους ήταν να βρουν μια απάντηση στα προβλήματα του παραδοσιακού προγραμματισμού, με τον προγραμματιστή να πρέπει να στήνει ένα περιβάλλον εργασίας κάθε φορά από την αρχή, κατεβάζοντας βιβλιοθήκες, ανεβάζοντας ξανά και ξανά σελίδες μέχρι την τελική έκδοση και έχοντας να αντιμετωπίσει αρκετά προβλήματα ρυθμίσεων στο περιβάλλον. Έτσι, άρχισαν την ανάπτυξη ενός διαδικτυακού εργαλείου ανάπτυξης εφαρμογών ιστού το οποίο προσφέρουν δωρεάν. Χρέωση υπάρχει μόνο για τις εταιρίες καθώς και για μεμονωμένους προγραμματιστές που απαιτούν προχωρημένες δυνατότητες και προκαλούν μεγαλύτερο φόρτο στο σύστημα. Αντίθετα, η ελεύθερη έκδοση παρέχει αρκετά για την ανάπτυξη μιας εφαρμογής ιστού. Ανάπτυξη που μπορεί να επιτευχθεί εκατό τοις εκατό μέσα από το περιβάλλον του kodingen.

Το kodingen, πλέον, αυτοπαρουσιάζεται ως "The Cloud Development Environment" και εκτός από ιστοσελίδα, διατίθεται και με τη μορφή εφαρμογής για το Chrome Web Store. Σύμφωνα με το ίδιο, πρόκειται για ένα cloud περιβάλλον ανάπτυξης που περιλαμβάνει συγγραφείς κώδικα, φιλοξενία εφαρμογών στο cloud, διαχείριση βάσεων δεδομένων ενώ συνεργάζεται με ένα web-based σύστημα αρχείων, με ftp servers και με subversion servers.

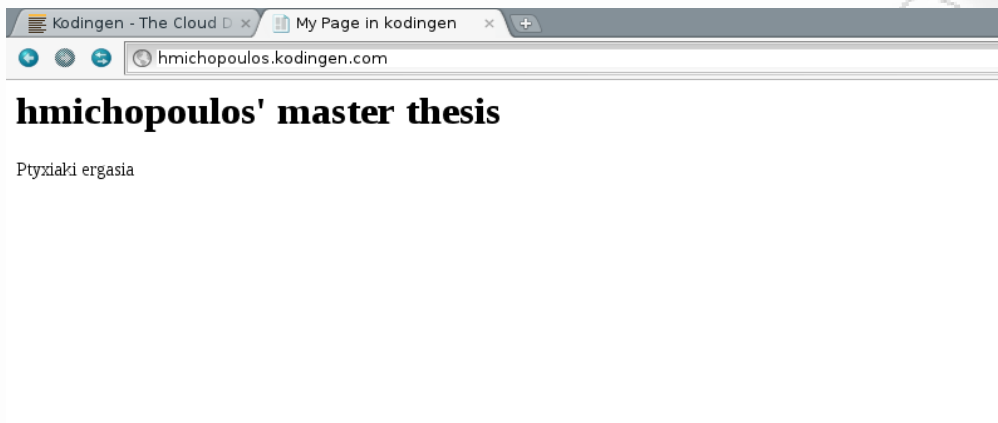
Παρέχει μια σειρά από συγγραφείς κώδικα, όπως το Bespin, Codemirror, ymacs κ.α. που υποστηρίζουν php, perl, python, django, ruby και ruby on rails, και άλλες. Μπορεί να συνδεθεί με εξωτερικούς ftp servers ενώ για τον έλεγχο των εκδόσεων του κώδικα παρέχει συνεργασία με git, subversion και mercurial.

Μέσα από το περιβάλλον εργασίας, μπορούμε να επεξεργαστούμε τα αρχεία μας με το εργαλείο της επιλογής μας, να εγκαταστήσουμε εφαρμογές και βιβλιοθήκες από μια σειρά έτοιμων ή να “ανεβάσουμε” τις δικές μας.

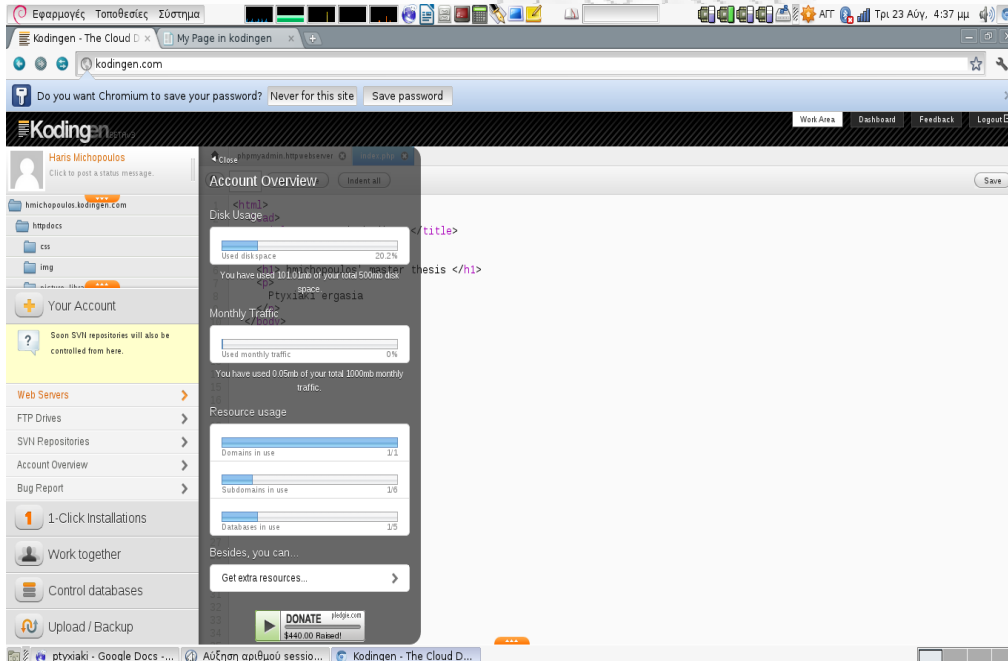
Μετά την εγγραφή μας και την ενεργοποίηση του λογαριασμού μας, μπορούμε να χρησιμοποιήσουμε απ’ ευθείας την ιστοσελίδα του kodingen ή να εγκαταστήσουμε το chrome app. Μπαίνοντας στη σελίδα ανάπτυξης, θα συναντήσουμε ένα περιβάλλον που μοιάζει περισσότερο με ένα απλό περιβάλλον ανάπτυξης παρά με τα υπερφορτωμένα πλέον κλασσικά εργαλεία ανάπτυξης.



Στο αριστερό μέρος βλέπουμε τα αρχεία του ιστοχώρου μας. Μπορούμε φυσικά να δημιουργήσουμε νέους φακέλους και αρχεία, να διαγράψουμε ή να μετονομάσουμε τα υφιστάμενα καθώς και να τα επεξεργαστούμε. Κάνοντας διπλό κλικ σε ένα αρχείο, εμφανίζεται ένα μενού με τους διαθέσιμους editors για να επιλέξουμε με ποιον θα δουλέψουμε. Έπειτα ανοίγει το αρχείο και οποιαδήποτε αλλαγή πραγματοποιούμε είναι άμεσα εμφανής στην εφαρμογή μας.

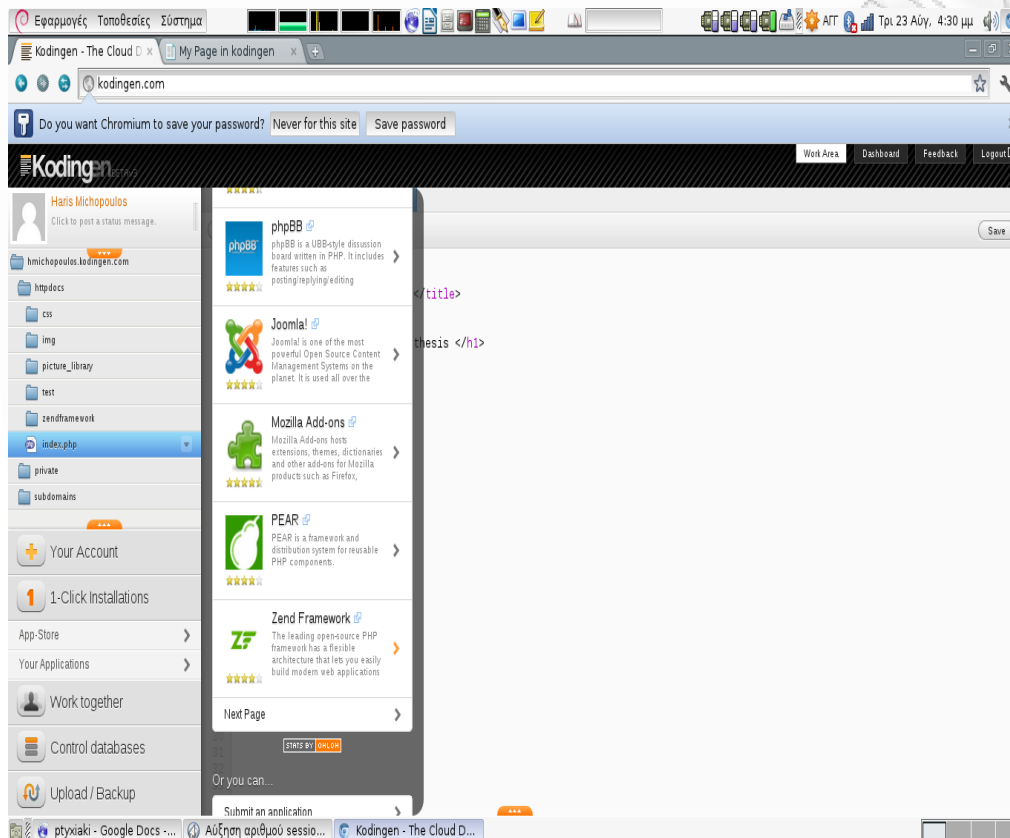


Κάτω από τα αρχεία του domain, θα βρούμε ένα μενού για λειτουργίες σχετικές με το λογαριασμό μας, όπως η δημιουργία νέου domain, η καταχώρηση web servers, ftp drivers και svn repositories ενώ υπάρχει και η επιλογή για προβολή προεπισκόπησης των στοιχείων του λογαριασμού μας.



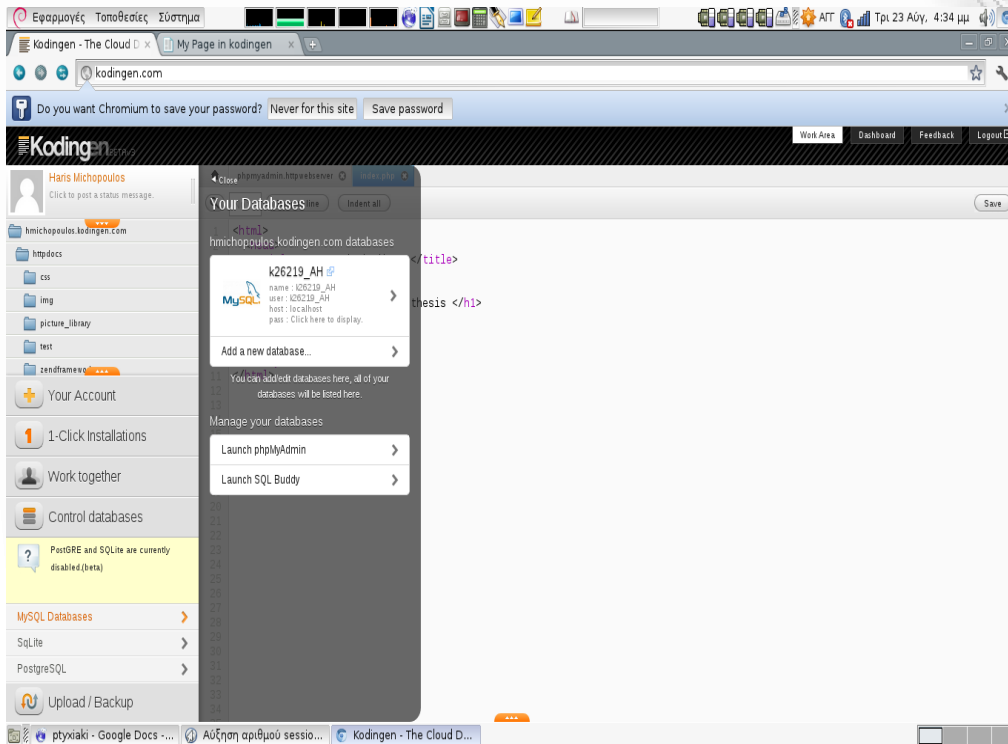
Συνεχίζοντας την εργασία μας με το kodingen παρατηρούμε το 1-click installations, ένα εργαλείο για την εγκατάσταση λογισμικού - εφαρμογών ή/και βιβλιοθηκών - στο domain που δουλεύουμε,

πχ μπορούμε να εγκαταστήσουμε το ZendFramework, ένα framework για php, ώστε να είναι διαθέσιμο πλέον για την ανάπτυξη της δικής μας εφαρμογής.



Πιο κάτω βλέπουμε τις δυνατότητες συνεργασίας που μας παρέχει το kodingen, με την υπολοίπη ομάδα μας. Προς το παρόν, οι δυνατότητες συγκρότησης ομάδας είναι απενεργοποιημένες.

Συνεχίζοντας, βρίσκουμε το μενού διαχείρισης βάσεων δεδομένων. Από εδώ μπορούμε να δημιουργήσουμε βάσεις δεδομένων - προς το παρόν, μόνο στο σύστημα mysql. Από εδώ μπορούμε επίσης να καλέσουμε το rhrmyadmin, μια εφαρμογή ιστού για διαχείριση βάσεων δεδομένων.



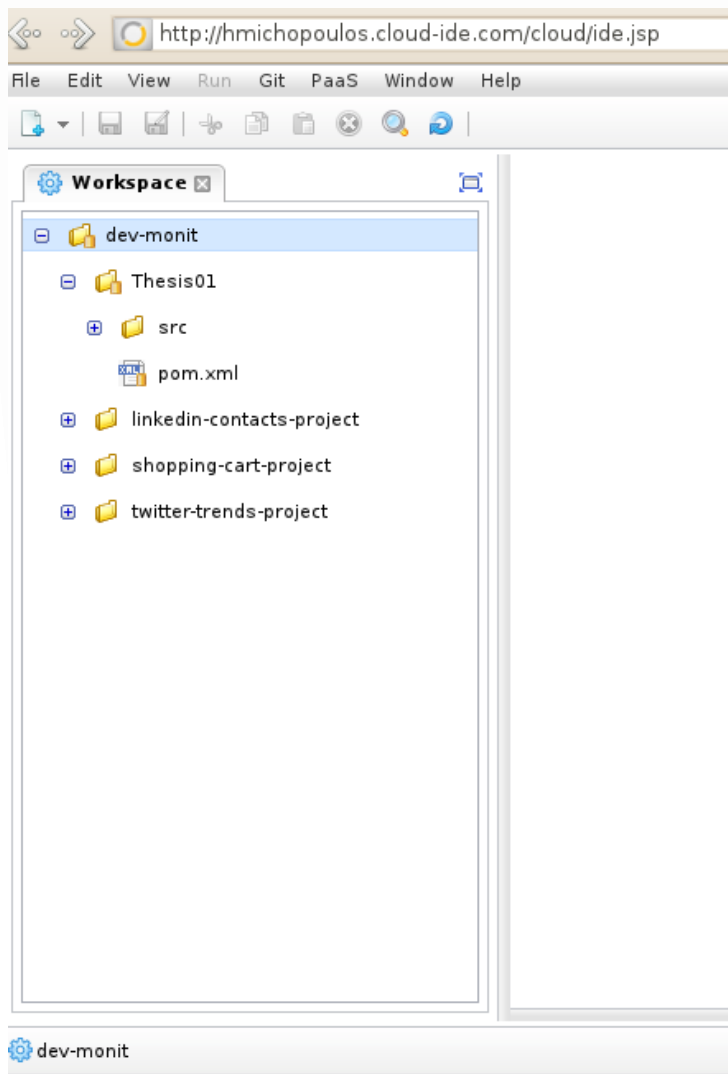
Στο τελευταίο μενού θα βρούμε επιλογές για “ανέβασμα” και “κατέβασμα” αρχείων από/προς τον υπολογιστή μας στο/από το server.

Cloudbees & could-ide.com

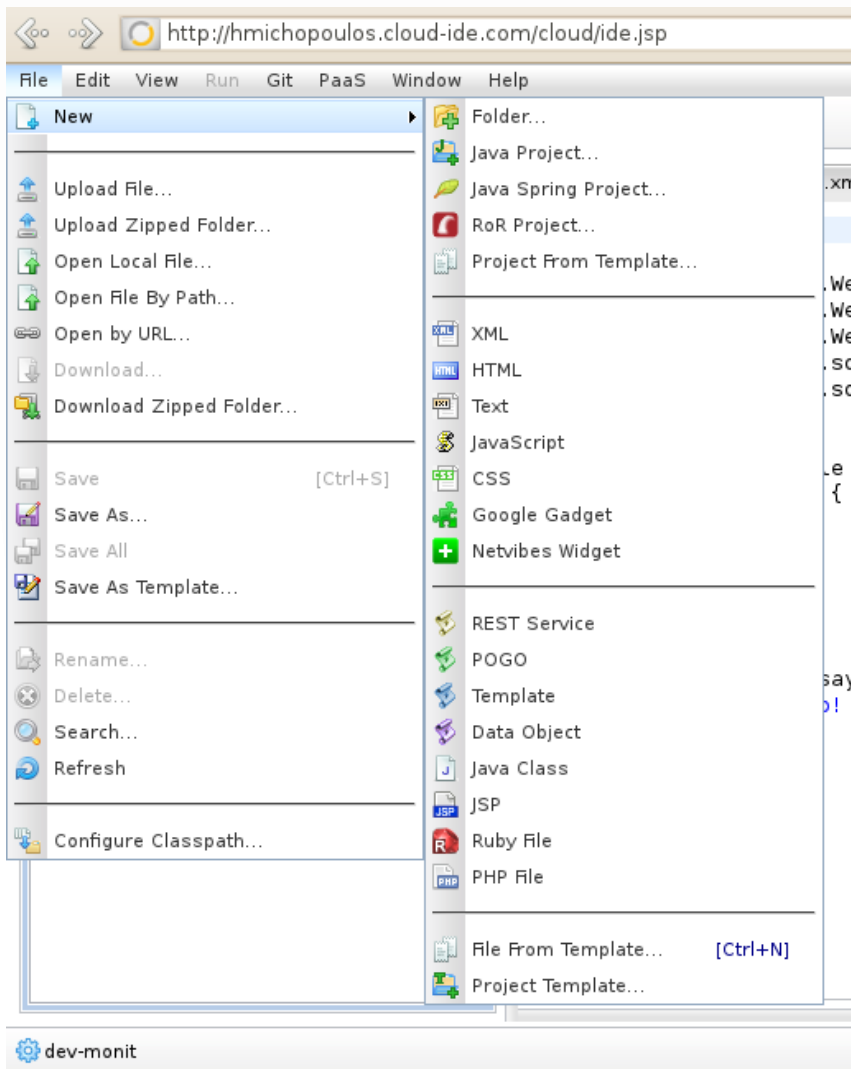
Η συγκεκριμένη πρόταση αποτελείται από δύο τμήματα. Και τα δύο λειτουργούν αυτόνομα, όμως συνεργάζονται. Το cloud-ide αποτελεί το περιβάλλον ανάπτυξης. Απευθύνεται σε προγραμματιστές java. Υποστηρίζει maven, subversion και git ενώ συνεργάζεται με τις πλατφόρμες cloudbees, cloudfoundry, heroku και openShift. Έτσι, παρέχει έμμεσα και τα εργαλεία της κάθε πλατφόρμας. Η πρώτη από αυτές, που θα χρησιμοποιήσουμε εδώ, προσφέρει εργαλεία continuous integration, testing κ.α. καθώς και την πλατφόρμα στην οποία θα τρέχουν οι εφαρμογές μας. Επιπλέον, το cloudbees παρέχει και πρόσθετα για το περιβάλλον ανάπτυξης eclipse (www.eclipse.org). Έτσι, θα κάνουμε και μια μικρή παρουσίαση για αυτή τη σύνθετη λύση, στην οποία έχουμε ένα παραδοσιακό περιβάλλον εργασίας με μια πλατφόρμα cloud computing.

Ξεκινώντας με το cloud-ide.com, παρατηρούμε ότι το περιβάλλον που μας παρουσιάζεται στην πρώτη σελίδα είναι αρκετά οικείο. Επάνω έχουμε ένα μενού όπως στις συνηθισμένες

εφαρμογές, λιγότερο φορτωμένο βέβαια, το οποίο μεταβάλλεται ανάλογα με το αρχείο το οποίο επεξεργαζόμαστε κάθε φορά. Ενώ, αριστερά βλέπουμε τη λίστα με τα έργα μας.



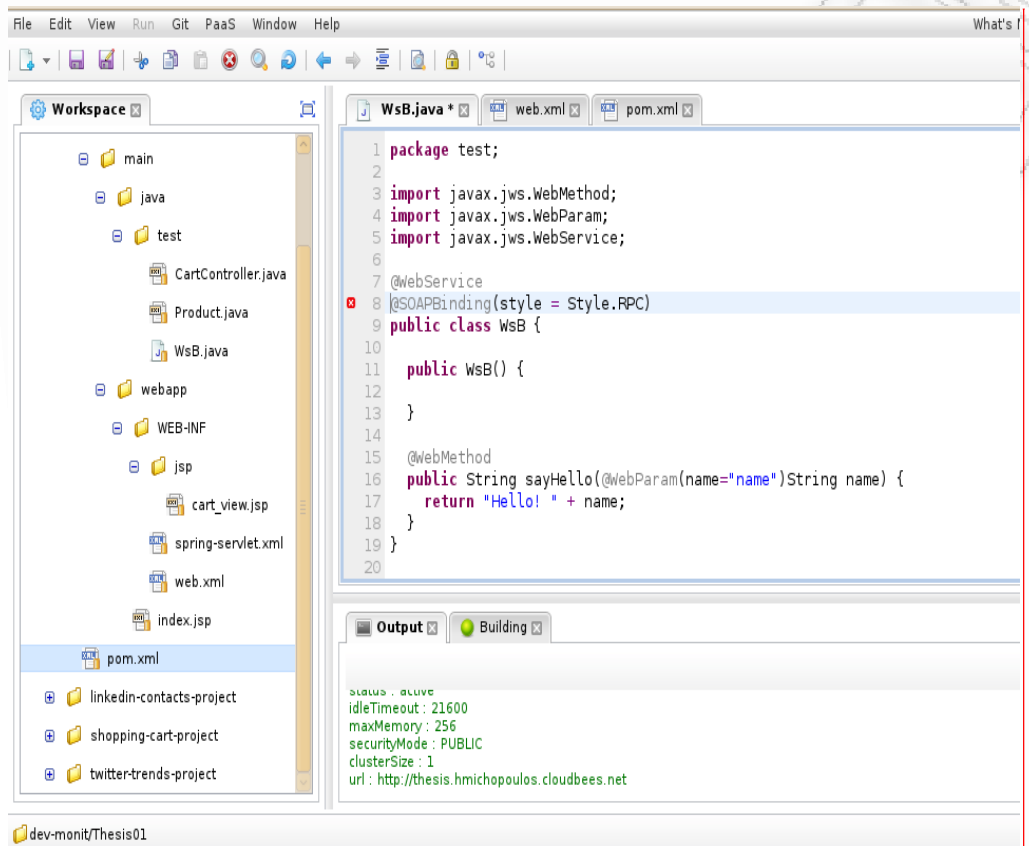
Το μενού του cloud-ide έχει τις αναμενόμενες λειτουργίες. πχ από το μενού "File->New" μπορούμε να δημιουργήσουμε ένα νέο έργο, επιλέγοντας από διάφορα πρότυπα, όπως java, spring mvc κ.α., ένα νέο αρχείο κλπ.



Από το μενού “Git” μπορούμε να διαχειριστούμε το versioning του κώδικά μας με το GIT, ένα κατανεμημένο σύστημα διαχείρισης εκδόσεων κώδικα, ενώ το σημαντικότερο μενού ίσως είναι το “PaaS” το οποίο διασυνδέει το cloud-ide με διάφορες πλατφόρμες cloud - μέχρι στιγμής με τις CloudBees, CloudFountry, Heroku και OpenShift.

Η δημιουργία ενός νέου έργου ακολουθεί την πεπατημένη. “File->New->Java Spring Project...”, πληκτρολογούμε το όνομα του έργου και πατάμε “Create”. Τότε, θα δημιουργηθεί ένα νέο maven-based project, το οποίο και θα εμφανιστεί στο αριστερό τμήμα της οθόνης. Από εκεί και μετά μπορούμε να αρχίσουμε να προσθέτουμε νέα αρχεία ή να επεξεργαζόμαστε τα υπάρχοντα.

Μπορούμε να πούμε ότι ο editor είναι αρκετά προσεγμένος και γρήγορος, εμφανίζει τα λάθη αλλά η αυτόματη συμπλήρωση που προσφέρει είναι ακόμα σε εμβρυικό στάδιο (πχ χρησιμοποιεί τα keywords της java, χωρίς όμως να λαμβάνει υπόψιν το context, τις μεθόδους τις κλάσεις κλπ)



Σχόλιο [2]: hmichopoulos:
Java editor with highlights and error-lines denoted

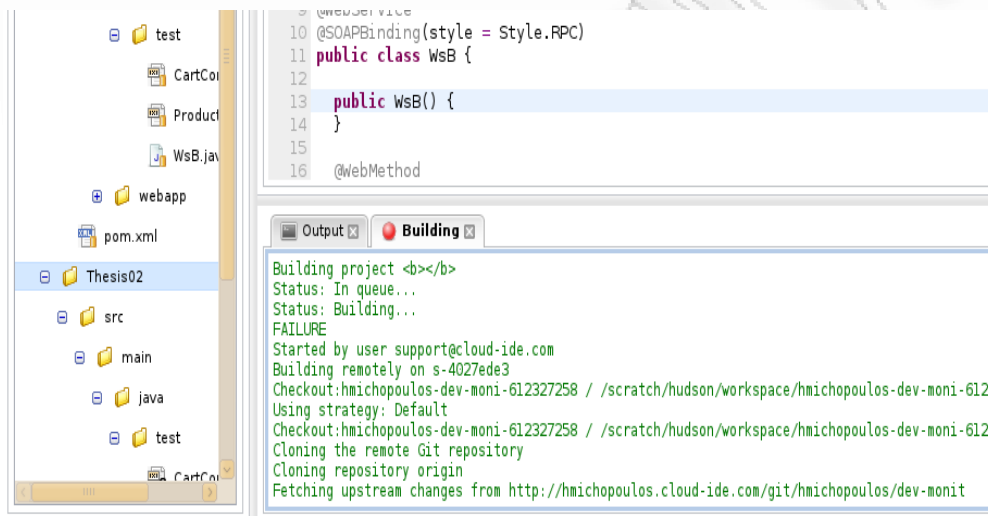
Στο παράδειγμα που βρίσκεται στο παράρτημα, με το όνομα Thesis01, έχουμε ένα Java Spring Project το οποίο αποτελείται από μια σελίδα και ένα web service, με το όνομα WsB.

Το ενδιαφέρον κομμάτι βέβαια εδώ, αφορά τη συνεργασία του cloud-ide με τους παρόχους cloud υπηρεσιών Platform as a Service. Το πρώτο πράγμα που έχουμε να κάνουμε είναι αρχικοποιήσουμε το git repository, πηγαίνοντας "Git->Initialize repository..." και συμπληρώνοντας κατάλληλα τη φόρμα που εμφανίζεται. Μετά, μπορούμε να προσθέσουμε το έργο μας στο repository, να προσθέσουμε τα αρχεία και να κάνουμε commit τις αλλαγές μας.

Το επόμενο βήμα είναι να κάνουμε push τον κώδικά μας στο remote repository του παρόχου PaaS με τον οποίο συνεργαζόμαστε. Αυτό σημαίνει πως θα πρέπει να έχουμε δημιουργήσει εκεί ένα repository, τις ρυθμίσεις του οποίου θα πρέπει να μεταφέρουμε και στο cloud-ide. Στην

περίπτωση του cloudbees πρέπει καταρχήν να δηλώσουμε τη διεύθυνσή του. Αυτό το κάνουμε πηγαίνοντας “Git->Remote->Remotes...”. Στην οθόνη που εμφανίζεται επιλέγουμε Add και συμπληρώνουμε ένα όνομα καθώς και τη διεύθυνση. Τη διεύθυνση μπορούμε να τη βρούμε στη σελίδα του cloudbees. Έπειτα, προκειμένου να επικοινωνεί το cloud-ide με το git repository στο cloudbees, πρέπει να δηλώσουμε ένα ssh κλειδί, στο “Windows->SSH Key Manager”. Στο παράθυρο που θα εμφανιστεί είτε φορτώνουμε ένα δικό μας κλειδί, το οποίο μπορεί να έχουμε δημιουργήσει στο pc μας, είτε πατάμε το Generate. Φυσικά το αντίστοιχο δημόσιο κλειδί πρέπει να το έχουμε δηλώσει στο cloudbees (Account->User settings).

Αφού ολοκληρώσουμε τη ρύθμιση του GIT και του remote repository, τότε περνάμε στο επόμενο στάδιο, αυτό του deployment της εφαρμογής. Πηγαίνουμε στο μενού “PaaS-> Cloudbees -> Initialize Application...”. Συμπληρώνουμε το όνομα της εφαρμογής και πατάμε “Create”. Στο σημείο αυτό θα αρχίσει το build, για το οποίο ενημερωνόμαστε στο αντίστοιχο παράθυρο.



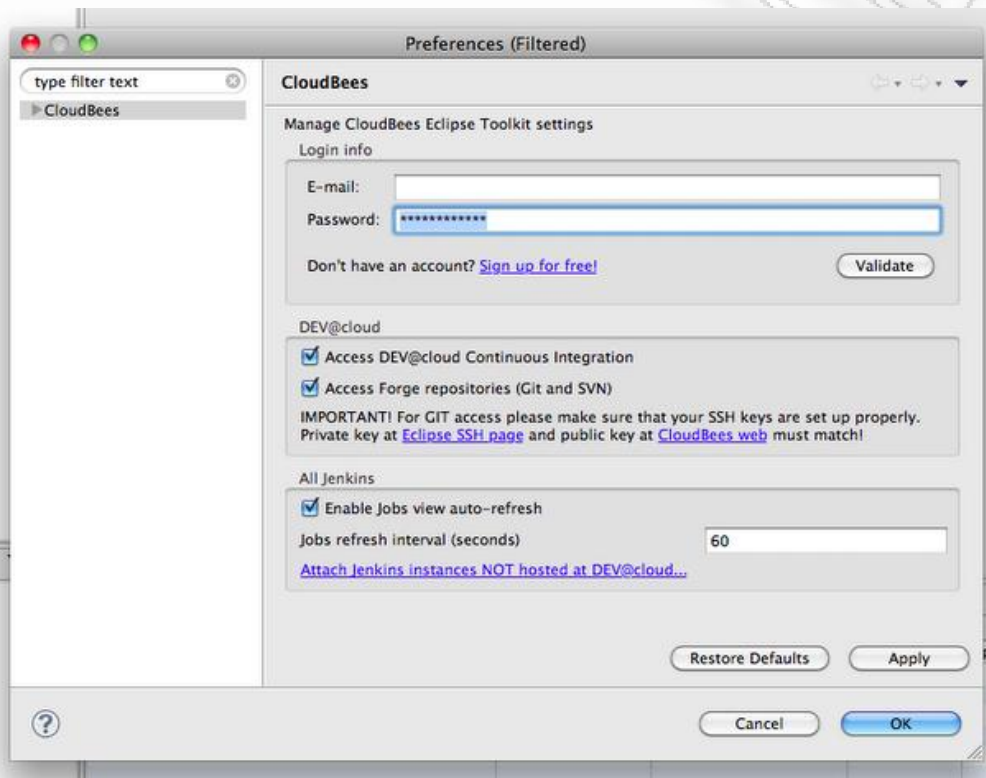
Όταν το build ολοκληρωθεί θα μας εμφανιστεί η διεύθυνση στην οποία μπορούμε να δούμε την εφαρμογή μας. Αν και αυτός ο τρόπος είναι αρκετά γρήγορος, ο πιο σωστός τρόπος είναι ρυθμίσουμε το build system στην πλατφόρμα μας ώστε να γίνεται εκεί το build, αφού έχουμε κάνει git push τον κώδικά μας. Αυτό, στο cloudbees γίνεται από το service Jenkins, έναν Continuous Integration server.

Ανάπτυξη με το eclipse

Αξίζει εδώ να αναφέρουμε και έναν συνδυασμό συγγραφής κώδικα με παραδοσιακό IDE με εκτέλεση στο cloud. Το IDE που θα χρησιμοποιήσουμε είναι το γνωστό eclipse (www.eclipse.com) ενώ η πλατφόρμα η cloudbees. Το cloudbees παρέχει ένα πρόσθετο για το eclipse το οποίο μας επιτρέπει να εργαζόμαστε τοπικά, να εκτελούμε την εφαρμογή μας τοπικά,

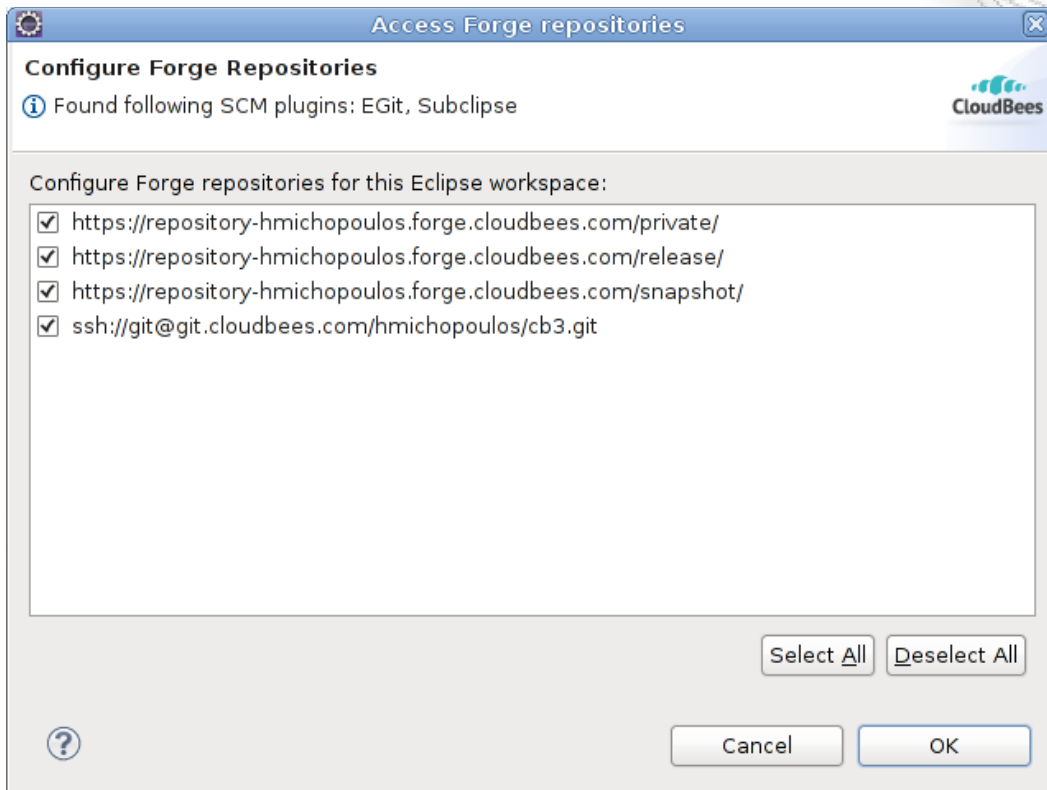
σε ένα περιβάλλον που προσομοιώνει το περιβάλλον της πλατφόρμας και φυσικά να χρησιμοποιούμε το git (σε συνδυασμό με το αντίστοιχο eGit plugin του eclipse) για τη διαχείριση των εκδόσεων του κώδικα.

Εκτελούμε λοιπόν το eclipse και εγκαθιστούμε το egit από το eclipse marketplace. Έπειτα εγκαθιστούμε το plugin του cloudbees από το <http://eclipse.cloudbees.com/> και κάνουμε restart το eclipse. Το πρώτο βήμα που πρέπει να κάνουμε είναι δηλώσουμε δηλώσουμε τα στοιχεία του cloudbees account μας στο eclipse, επιλέγοντας Windows->Preferences->Cloudbees. Συμπληρώνουμε τα στοιχεία μας και πατάμε validate.



Επιπλέον, εάν το ssh κλειδί που χρησιμοποιούμε για το git στο cloudbees δεν το έχουμε παράξει στο ίδιο μηχάνημα, τότε πρέπει να το προσθέσουμε στο ssh.

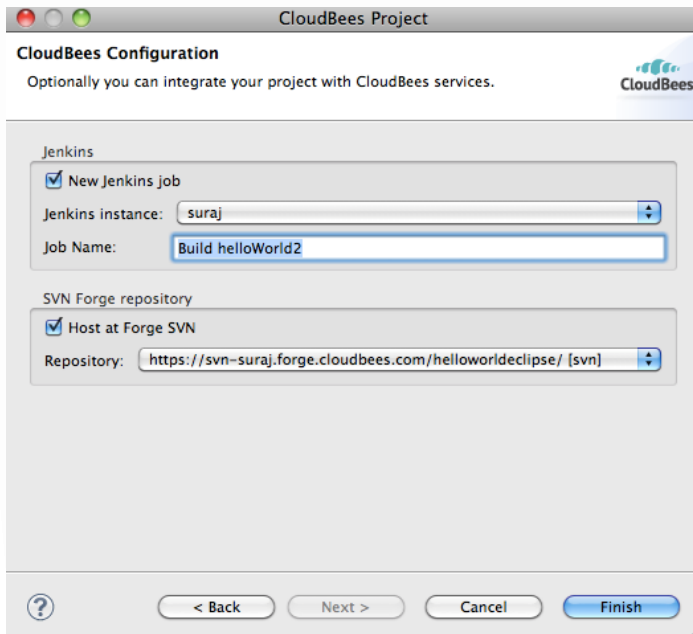
Ανοίγοντας το cloudbees perspective, από το Windows->Open perspective->Others->Cloudbees, θα μας εμφανιστεί ένα παράθυρο για τη ρύθμιση των repositories που έχουμε στο cloudbees.



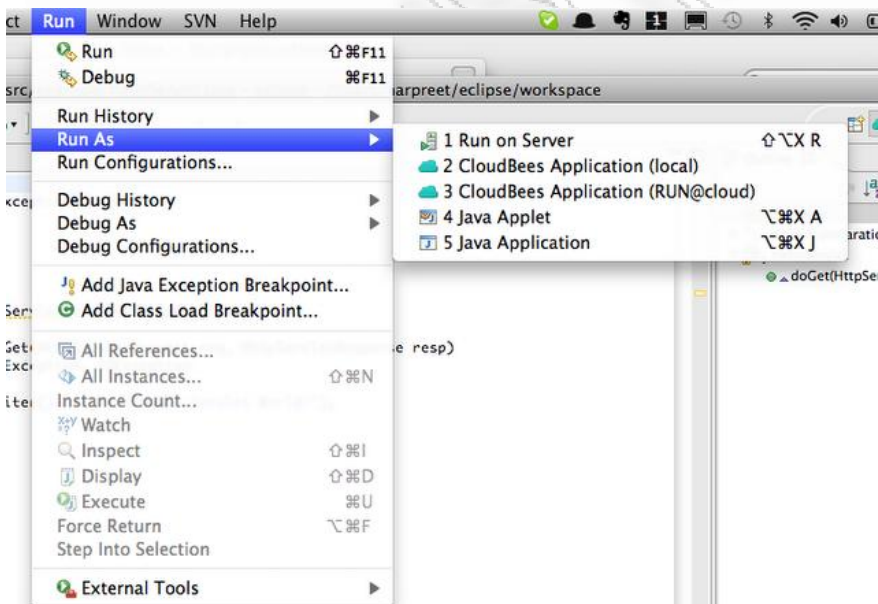
Επιλέγουμε αυτό στο οποίο δουλεύουμε και πατάμε οκ.

Για να συνεργαστεί το eclipse με το cloudbees πρέπει να έχουμε ένα subversion repository στο τελευταίο.

Έπειτα δημιουργούμε ένα νέο cloudbees project. Στο τελευταίο βήμα δηλώνουμε ένα jenkins job να κάνει build το έργο μας και το subversion repository που θα διαχειρίζεται τον κώδικά μας.



Αφού εκτελέσουμε τοπικά την εφαρμογή μας και ελέγξουμε την ορθότητά της, την κάνουμε commit στο subversion ούτως ώστε να γίνει το build από το jenkins.



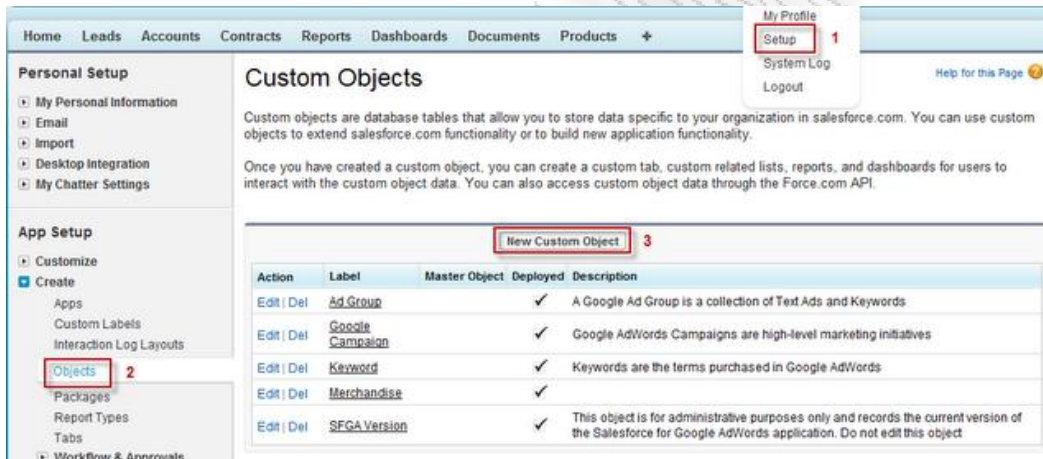
Είδαμε λοιπόν εν συντομία και μία λύση που συνδυάζει την ανάπτυξη σε παραδοσιακό περιβάλλον με την εκτέλεση σε cloud πλατφόρμα.

Salesforce.com

Το [salesforce.com](https://www.salesforce.com) έγινε γνωστό για τις CRM λύσεις που παρείχε. Πλέον, προχωρά ένα βήμα παραπέρα παρέχοντας την πλατφόρμα του ως βάση για την ανάπτυξη εφαρμογών. Ταυτόχρονα, προσφέρει ένα περιβάλλον ανάπτυξης εφαρμογών καθώς και το περιβάλλον εκτέλεσής τους.

Αν ακολουθήσουμε τις οδηγίες του [salesforce.com](https://www.salesforce.com) για την δημιουργία μιας εφαρμογής θα έρθουμε μπροστά σε εικόνες οικείες σε αυτούς που υλοποιούν λύσεις CRM ή ERP.

Δημιουργούμε ένα custom object, προσθέτουμε χαρακτηριστικά, ορίζουμε τις ιδιότητές τους. Η διαδικασία είναι αρκετά όμοια με αυτή που ακολουθείται σε παραδοσιακά CRM όπως το SugarCRM, το tigerCRM κ.α.



The screenshot displays the Salesforce interface for managing Custom Objects. The left sidebar shows the 'App Setup' menu with 'Objects' selected. The main content area is titled 'Custom Objects' and includes a table of existing objects. A 'New Custom Object' button is visible at the top of the table. The 'My Profile' menu is also visible in the top right corner.

Action	Label	Master Object	Deployed	Description
Edit Del	Ad Group		✓	A Google Ad Group is a collection of Text Ads and Keywords
Edit Del	Google Campaign		✓	Google AdWords Campaigns are high-level marketing initiatives
Edit Del	Keyword		✓	Keywords are the terms purchased in Google AdWords
Edit Del	Merchandise		✓	
Edit Del	SFGA Version		✓	This object is for administrative purposes only and records the current version of the Salesforce for Google AdWords application. Do not edit this object

Custom Object Definition Edit Save Save & New Cancel

Custom Object Information Required Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label **Example: Account**

Plural Label **Example: Accounts**

Starts with vowel sound

The Object Name is used when referencing the object via the API.

Object Name **Example: Account**

Description

Context-Sensitive Help Setting Open the standard Salesforce.com Help & Training window Open a window using a Visualforce page

Content Name

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and search results. For example, the Record Name for Account is "Account Name" and for Case it is "Case Number". Note that the Record Name field is always called "Name" when referenced via the API.

Record Name **Example: Account Name**

Data Type

Custom Object Help for this Page

Merchandise

[Standard Fields \(4\)](#) | [Custom Fields & Relationships \(0\)](#) | [Validation Rules \(0\)](#) | [Page Layouts \(1\)](#) | [Search Layouts \(4\)](#) | [Standard Buttons and Links \(8\)](#) | [Custom Buttons and Links \(0\)](#) | [Record Types \(0\)](#) | [Apex Sharing Reasons \(0\)](#) | [Apex Sharing Recalculation \(0\)](#)

Custom Object Definition Detail Edit Delete

Singular Label	Merchandise	Description	
Plural Label	Merchandise	Enable Reports	<input type="checkbox"/>
Object Name	Merchandise	Track Activities	<input type="checkbox"/>
API Name	Merchandise__c	Track Field History	<input type="checkbox"/>
		Deployment Status	Deployed
		Help Settings	Standard salesforce.com Help Window
Created By	Liz Garcia, 10/1/2010 4:32 PM	Modified By	Liz Garcia, 10/1/2010 4:32 PM

Standard Fields Standard Fields Help

Action	Field Label	Field Name	Data Type	Controlling Field
	Created By	CreatedBy	Lookup(User)	
	Last Modified By	LastModifiedBy	Lookup(User)	
Edit	Merchandise Name	Name	Text(80)	
Edit	Owner	Owner	Lookup(User.Queue)	

Custom Fields & Relationships New Field Dependencies Custom Fields & Relationships Help

No custom fields defined

Αυτό που ξεχωρίζει το salesforce από τις προηγούμενες λύσεις είναι ότι προσφέρει λειτουργίες που παρέχει ήδη από την SaaS λύση του για τη δημιουργία συγκεκριμένων εφαρμογών τύπου CRM. Με άλλα λόγια, διαθέτει την ήδη υπάρχουσα τεχνογνωσία και υποδομή που έχει αναπτυχθεί κατά την ανάπτυξη της Software as a Service CRM λύσης του, ως πλατφόρμα για την ανάπτυξη παρόμοιων εφαρμογών.

Συμπεράσματα

Απ' ότι φαίνεται τα περιβάλλοντα ανάπτυξης παρασύρονται κι αυτά από τη δυναμική του cloud computing μεταφερόμενα στο διαδίκτυο. Πλέον, υπάρχουν αρκετές λύσεις οι οποίες βελτιώνονται μέρα με την ημέρα ενώ συνεχώς εμφανίζονται νέες. Βέβαια υπολείπονται κατά πολύ από το να χαρακτηριστούν ολοκληρωμένα, αφού δεν έχουν βασικές λειτουργίες όπως αυτόματη συμπλήρωση, compilation, ενοποίηση με bug tracking συστήματα, υποδείξεις κ.α. - χαρακτηριστικά που θεωρούνται αυτονόητα στα σύγχρονα IDEs.

Βέβαια, κάποια κομμάτια όπως τα version control systems και οι CI servers έχουν ήδη μετακομίσει σε πλατφόρμες cloud και αναμένεται να συντελέσουν τα μέγιστα στην επιτάχυνση της ανάπτυξης βοηθώντας κυρίως το QA και το UAT. Στο επόμενο κεφάλαιο θα δούμε κάποιες πλατφόρμες cloud.

Infrastructure as a Service

Θα περιγράψουμε τώρα τις βασικές υπηρεσίες που πρέπει να παρέχονται στο επίπεδο των υποδομών. Έπειτα θα παρουσιάσουμε το βασικότερο πάροχο υπηρεσιών υποδομής, οποίος μάλιστα θεωρείται από τους πρωτοπόρους και παρέχει ίσως το πληρέστερο σύνολο υπηρεσιών υποδομής.

Βασικές υπηρεσίες υποδομής

Από τον όρο υποδομή μπορούμε να αντιληφθούμε ότι ένας πάροχος τέτοιων υπηρεσιών οφείλει να προσφέρει υπολογιστικούς πόρους σε πολύ χαμηλό επίπεδο. Όντως, τα βασικά προϊόντα που προσφέρονται είναι: αποθηκευτικός χώρος (storage), επεξεργαστική ισχύς και μνήμη. Μολονότι αυτά δεν ακούγεται ως κάτι το ιδιαίτερο θα πρέπει να εξετάσουμε υπό το πρίσμα των απαιτήσεων του cloud computing. Εκεί θα αντιληφθούμε και τις διαφορές που έχει ενά παραδοσιακό σύστημα σε σχέση με το cloud.

CPU and RAM

Το cloud βασίζεται στην τεχνολογία virtualization και το βασικό αγαθό που λαμβάνει ένας πελάτης είναι ένα virtual machine. Μέχρι στιγμής αυτά είναι διαθέσιμα και σε ένα παραδοσιακό data center. Καιρός λοιπόν να βάλουμε και τις απαιτήσεις του cloud computing στο χώρο. Η πρώτη απαίτηση είναι η ελαστικότητα. Εάν ενοικιάσουμε ένα server, φυσικό ή virtual, σε ένα παραδοσιακό data center, για να βελτιώσουμε τα χαρακτηριστικά του θα πρέπει να τον αντικαταστήσουμε. Αντίθετα, στο cloud μπορούμε να κάνουμε επανεκκίνηση του instance σε ένα virtual machine με διαφορετικά χαρακτηριστικά. Επιπλέον, μπορούμε άμεσα να εκκινήσουμε όσους virtual servers χρειαζόμαστε από το ίδιο image.

Τα images ουσιαστικά αποτελούν πρότυπα, εικόνες λειτουργικών συστημάτων μαζί με κάποιο λογισμικό. Ο χρήστης του cloud εκκινεί ένα virtual machine instance χρησιμοποιώντας κάποιο image, είτε δικό του, είτε κάποιο από τα δημοσίως διαθέσιμα. Τα πρότυπα, όμως, αυτά είναι αμετάβλητα, αλλιώς δεν θα είχαν νόημα αν ο καθένας που χρησιμοποιούσε ένα πρότυπο προσπαθούσε λογισμικό. Αυτό δημιουργεί ένα σημαντικό πρόβλημα, οι αλλαγές που πραγματοποιούνται στο instance δεν αποθηκεύονται στο αντίστοιχο image. Με άλλα λόγια, το σύστημα αρχείων του κάθε instance είναι προσωρινό και για όσο διαρκεί η εκτέλεσή του. Αυτό οδήγησε στην παροχή υπηρεσιών μόνιμου αποθηκευτικού χώρου αλλά και την ανάγκη παροχής υπηρεσιών αποθήκευσης δεδομένων, όπως databases, queues ή logging. Αν και αυτά

μπορούν να θεωρηθούν υπηρεσίες πλατφόρμας, είναι τόσο βασικά στοιχεία για κάθε εφαρμογή που παρέχονται συνήθως και στο επίπεδο των υπηρεσιών υποδομής.

Επιπλέον, μια άλλη διαφοροποίηση του cloud από τα παραδοσιακά datacenters έχει να κάνει με το billing και το metering. Ενώ στα παραδοσιακά datacenters η χρέωση είναι σταθερή, πχ ανά μήνα, στο cloud η χρέωση γίνεται πάντα με βάση τη χρήση. Επομένως, απαιτούνται συστήματα μέτρησης των ωρών χρήσης σε συνδυασμό με τα χαρακτηριστικά του κάθε virtual machine instance ώστε να βγαίνουν ακριβείς και σωστές χρεώσεις.

Storage

Πολλές εταιρίες προσέφεραν αποθηκευτικό χώρο στο διαδίκτυο. Ο χώρος αυτός είναι προσβάσιμος μέσω ειδικών προγραμμάτων στους χρήστες. Μια τέτοια γνωστή υπηρεσία παρέχεται από το Dropbox. Ο χρήστης με το πρόγραμμα της ομώνυμης εταιρίας, μπορεί να διατηρεί αντίγραφα κάποιων αρχείων του στο διαδίκτυο και να τα συγχρονίζει με διάφορους υπολογιστές. Έτσι, τα αρχεία του είναι διαθέσιμα και ενημερωμένα τόσο στους υπολογιστές που χρησιμοποιεί όσο και on-line στην αντίστοιχη ιστοσελίδα.

Το πρόβλημα των προσωρινών συστημάτων αρχείων των virtual machines οδήγησε στην εξέλιξη αυτής της υπηρεσίας με τον εξής τρόπο. Αντί να παρέχουν κάποιον αποθηκευτικό χώρο σε υπάρχον σύστημα αρχείων, όπως το Dropbox, με την εταιρία να διαχειρίζεται το σύστημα αρχείων και να είναι υπεύθυνη γι' αυτό, προσφέρουν πλέον "μέρος του δίσκου" χωρίς μορφοποίηση για αποθήκευση. Ο χρήστης του cloud μπορεί να προσαρτήσει αυτό το κομμάτι δίσκου που έχει ενοικιάσει σε ένα virtual instance και να το μορφοποιήσει και να το χρησιμοποιεί σαν να πρόκειται για ένα παραδοσιακό partition.

Databases

Η υπηρεσία αυτή θα μπορούσε κάλλιστα να θεωρηθεί ότι ανήκει στο επίπεδο της πλατφόρμας. Λόγω όμως της σημασίας των βάσεων δεδομένων σε όλες σχεδόν τις εφαρμογές αλλά και λόγω της αφαίρεσης των ιδιαιτεροτήτων κάθε RDBMS που προσφέρουν οι drivers, όπως jdbc, η υπηρεσία αυτή παρέχεται στο επίπεδο της υποδομής.

Με την υπηρεσία αυτή, παρέχονται στους χρήστες του cloud βάσεις δεδομένων κατ' απαίτηση. Δηλαδή, κάθε χρήστης μπορεί να δημιουργεί βάσεις, σχήματα (database schemas), πίνακες, χρήστες κλπ και να τα χρησιμοποιεί για όσο τα χρειάζεται. Έτσι, αφ' ενός ξεπερνά το πρόβλημα της μόνιμης αποθήκευσης των δεδομένων του και αφ' εταίρου επωφελείται από την ελαστικότητα που προσφέρει το cloud. Επιπλέον, λόγω των οικονομιών κλιμακας του παρόχου, μπορεί να χρησιμοποιήσει ακόμα και εμπορικά συστήματα βάσεων δεδομένων πληρώνοντας ως ενοίκιο ελάχιστα σε σχέση με αυτά που θα έπρεπε να πληρώσει για την αγορά του μεγιστοποιώντας το ROI.

Πρέπει να σημειωθεί εδώ και ο ρόλος που παίζουν τα νέα συστήματα βάσεων δεδομένων που δεν ακολουθούν τις αρχές των σχεσιακών συστημάτων και αυτοχαρακτηρίζονται ως noSQL Database Management systems. Τα συστήματα αυτά, είτε καλύπτουν εξειδικευμένες ανάγκες, όπως ταχύτητα πχ Kyoto Cabinet/ Kyoto Tycoon, είτε προσφέρουν χαλαρά σχήματα (soft-schema) όπως το cassandra και στοχεύουν σε κατανεμημένα συστήματα όπως το Hadoop, το Google Map/Reduce ή το Voldermart και το KumoFS.

Φυσικά, αυτά τα συστήματα ταιριάζουν "γάντι" με τις απαιτήσεις του cloud computing για scaling, αφού έχουν σχεδιαστεί ώστε να είναι κατανεμημένα. Έτσι, και αυτά τα DBS πλέον γίνονται μέρος των υπηρεσιών υποδομής με ταχύς ρυθμούς.

Queues

Μία άλλη υπηρεσία που θεωρητικά θα έπρεπε να ανήκει στο ανωτέρω επίπεδο είναι αυτό των message broker queues. Πρόκειται για συστήματα τα οποία χρησιμοποιούνται για την ασύγχρονη επικοινωνία δύο ή περισσότερων συστημάτων μέσω της αποστολής και λήψης μηνυμάτων προς και από μια ουρά μηνυμάτων.

Αυτή η τεχνολογία σήμερα έχει γίνει ζωτικής σημασίας για τις περισσότερες εφαρμογές αφού συμβάλει αποφασιστικά στην κατανομή της εργασίας. Για παράδειγμα, ένα σύστημα μπορεί να στέλνει μηνύματα σε μία ουρά, αλλά από την ουρά αυτή να διαβάζουν ένα ή περισσότερα συστήματα και να επεργάζονται τα μηνύματα. Τα συστήματα επεξεργασίας μπορούν να αυξομειώνονται κατ' ανάγκη εφαρμόζοντας τέλεια την αρχή της ελαστικότητας.

Amazon's Elastic Compute Cloud (EC2)

Όπως έχουμε ήδη αναφέρει, η Amazon ήταν από τους πρωτοπόρους στο χώρο και παρέχει σήμερα μια από τις πληρέστερες υπηρεσίες υποδομής. Ταυτόχρονα, έχει αρχίσει να προσφέρει και κάποιες υπηρεσίες που μπορούν να θεωρηθούν και υπηρεσίες πλατφόρμας.

Λόγω του ρόλου που έχει επιτελέσει στο cloud computing αλλά και του γεγονότος ότι είναι το de facto standard (ενδεικτικό αυτού είναι πως οι νέοι πάροχοι φροντίζουν να είναι συμβατοί με το σύστημα της Amazon) αξίζει να γίνει μια αναφορά στην υπηρεσία της και σε αυτά που προσφέρει.

Υπηρεσίες του Amazon's EC2

AWS Elastic Beanstalk

Πρόκειται για μια υπηρεσία με στόχο την εύκολη και γρήγορη διαχείριση των εφαρμογών που τρέχουν στο cloud και το deployment αυτών. Το Elastic Beanstalk αναλαμβάνει όλες τις λεπτομέρειες του deployment όπως η παροχή χωρητικότητας, το load balancing, αυτόματη κλιμάκωση (auto-scaling) και την παρακολούθηση της ευρωστίας της εφαρμογής. Ταυτόχρονα, οι εφαρμογές που τρέχουν στο Elastic Beanstalk έχουν πρόσβαση και στις υπόλοιπες υπηρεσίες του Amazon's EC2.

Μέχρι στιγμής προσφέρεται για εκτέλεση εφαρμογών Java χρησιμοποιώντας τον Apache Tomcat Application server. Επίσης, δεν υπάρχει ξεχωριστή χρέωση για το Beanstalk παρά μόνο για τη χρήση των πόρων που χρησιμοποιεί η εφαρμογή μας.

Amazon S3

Πρόκειται για υπηρεσία διαδικτυακής αποθήκευσης αρχείων και απευθύνεται στους προγραμματιστές υποσχόμενη να τους διευκολύνει σχετικά το web-scale computing.

Μέσω μιας ιστοσελίδας, ο χρήστης μπορεί να αποθηκεύσει, κατεβάσει ή διαγράψει τα αρχεία του από οποιοδήποτε σημείο στον κόσμο. Παρέχει στους προγραμματιστές την ίδια αξιόπιστη, γρήγορη και ασφαλή υποδομή που χρησιμοποιεί και η ίδια η Amazon. Μερικά από τα βασικά χαρακτηριστικά του είναι τα εξής:

- Κάθε αντικείμενο αποθηκεύεται σε ένα συγκεκριμένο bucket και είναι διαθέσιμο μόνο με ένα συγκεκριμένο κλειδί.
- Κάθε bucket μπορεί να βρίσκεται σε μία γεωγραφική περιοχή.
- Κανένα αντικείμενο δεν αλλάζει γεωγραφική περιοχή αυτόματα. Μόνο ο χρήστης μπορεί να το κάνει αυτό.
- Παρέχει έναν ευέλικτο μηχανισμό ασφαλείας και αδειοδότησης.
- Παρέχει SOAP και REST διεπαφές για τη διαχείριση των αντικειμένων
- Προσφέρεται με SLA (Service Level Agreement)

Αναφορικά με την προστασία των δεδομένων, το Amazon S3 παρέχει τα εξής:

- Πολλαπλά αντίγραφα των δεδομένων
- Ανοχή ακόμα κι αν χαθούν ταυτόχρονα δύο αντίγραφα
- Συνεχή έλεγχο με χρήση checksums και αυτόματη επιδιόρθωση εάν ανιχνευτεί πρόβλημα
- Χρήση checksums κατά τη μεταφορά για την αποφυγή λαθών
- Versioning των αρχείων, ούτως ώστε να διατηρείται η ιστορικότητά τους και να είναι εφικτή η επαναφορά τους.

Επίσης, προσφέρει και χώρο μειωμένης ανοχής σε προβλήματα (Reduced Redundancy Storage, RRS) για αποθήκευση λιγότερο κρίσιμων αρχείων με χαμηλότερο φυσικά κόστος.

Amazon VPC

Το Amazon Virtual Private Cloud είναι μια υπηρεσία που μας εφοδιάζει με ένα ιδιωτικό, απομονωμένο κομμάτι από το Cloud της Amazon, στο οποίο μπορούμε να χρησιμοποιήσουμε τους διατιθέμενους πόρους δημιουργώντας ένα ιδιωτικό cloud. Μας παρέχει πλήρη έλεγχο πάνω στις ιδιότητες του δικτύου, όπως IP διευθύνσεις, δημιουργία υποδικτύων, ρύθμιση firewalls και router tables. Μπορούμε, για παράδειγμα, να δημιουργήσουμε ένα υποδίκτυο με δημόσια πρόσβαση για τις ιστοσελίδες μας και να τοποθετήσουμε τα πιο κρίσιμα τμήματα των εφαρμογών όπως βάσεις δεδομένων σε ένα υποδίκτυο χωρίς δημόσια πρόσβαση.

Επίσης, μπορούμε να δημιουργήσουμε ένα Hardware Virtual Private Network μεταξύ του εσωτερικού, δικού μας δικτύου, πχ το δίκτυο της εταιρίας μας και του Virtual Private Cloud μας ώστε το τελευταίο να λειτουργεί ως προέκταση του εσωτερικού μας δικτύου.

Amazon CloudWatch

Το Amazon CloudWatch παρέχει μηχανισμούς παρακολούθησης της χρήσης των πόρων του cloud από εμάς και τις εφαρμογές μας. Οι προγραμματιστές και οι διαχειριστές μπορούν να το χρησιμοποιούν για τον έλεγχο της λειτουργίας των εφαρμογών τους ώστε να λαμβάνουν άμεσα μέτρα σε περίπτωση που απαιτηθεί.

Είναι μια πολύ εύκολη λύση για την παρακολούθηση των συστημάτων που τρέχουν στο Amazon's EC2 και παρέχει μηχανισμούς για την παρακολούθηση των instances, volumes, Load Balancers, SQS topics, SNS topics και RDS databases σε πραγματικό χρόνο. Επιπλέον, μπορεί να επεκταθεί με την προσθήκη μετρητών για τα χαρακτηριστικά των εφαρμογών μας, όπως transaction volumes, μνήμη κλπ.

Elastic Load Balancing

Ο μηχανισμός Elastic Load Balancing χρησιμοποιείται για να διανείμει το φόρτο εργασίας αυτόματα στα amazon instances. Έτσι οι εφαρμογές γίνονται πιο αποκρίσιμες και αξιόπιστες. Ο μηχανισμός αυτός είναι επίσης σε θέση να διακρίνει instances τα οποία παρουσιάζουν προβλήματα σταθερότητας και να μεταφέρει το φόρτο εργασίας σε άλλα instances πιο εύρωστα.

Τα χαρακτηριστικά του μηχανισμού αυτού είναι τα εξής:

- Διανείμει αυτόματα τα incoming requests
- Προβαίνει σε αυτόματο scaling ενεργοποιώντας νέα instances όταν απαιτείται
- Κάνει διάγνωση της σταθερότητας των instances απομονώνοντας τα ασταθή.
- Παρέχει τη δυνατότητα να κατευθύνει σταθερά έναν χρήστη της εφαρμογής στο ίδιο instance.
- Παρέχει κεντρική διαχείριση των πιστοποιητικών SSL και επιτρέπει τη χρήση SSL μέχρι το load balancer αποφορτίζοντας τους application servers από την αποκρυπτογράφηση.

- Χρησιμοποιεί τόσο IPv4 όσο και IPv6.
- Παρέχει μετρήσεις για τους χρόνους απόκρισης και τον αριθμό των requests μέσα από το Amazon CloudWatch.

Elastic Block Store (ESB)

Το Elastic Block Store παρέχει αποθηκευτικό χώρο (volumes) σε επίπεδο block (σχεδόν συσκευής, δηλαδή) για χρήση από τα amazon instances. Τα δεδομένα στο ESB volume διατηρούνται ανεξάρτητα από τα δεδομένα του instance. Ουσιαστικά παρέχει αποθηκευτικό χώρο που μπορεί να προσαρτηθεί σε ένα instance και διατεθεί σε αυτό με τη μορφή συσκευής. Το instance θεωρεί ότι απλώς έχει προστεθεί ένας σκληρός δίσκος στο σύστημα. Ο μηχανισμός αυτός είναι απαραίτητος για τη μόνιμη αποθήκευση δεδομένων μέσω των εφαρμογών, πχ logging.

Οι δυνατότητες που προσφέρει αυτός ο μηχανισμός συνοψίζονται στις εξής:

- Προσφέρει χώρους από 1GB έως 1TB.
- Κάθε instance μπορεί να χρησιμοποιεί παραπάνω από ένα ESB volume.
- Τα ESB volumes είναι διαθέσιμα στο σύστημα σαν raw, unformatted devices.
- Προσφέρει αυτόματο replication των volumes
- Παρέχει τη δυνατότητα να λαμβάνονται snapshots τα οποία αποθηκεύονται στο ESB και μπορούν να χρησιμοποιούνται κανονικά ως volumes
- Παρέχει μετρήσεις της απόδοσης και της ταχύτητας των volumes μέσα από το Amazon CloudWatch.

Amazon CloudFront

Πρόκειται για μια δικτυακή υπηρεσία παράδοσης περιεχομένου (content delivery). Το περιεχόμενο μπορεί να είναι multimedia, αρχεία, streaming κλπ. Η υπηρεσία είναι ενοποιημένη με τις υπόλοιπες και προσφέρει στους προγραμματιστές ένα διαφανή και εύκολο τρόπο να προσφέρουν μέσα από τις εφαρμογές τους περιεχόμενο σε υψηλής ταχύτητας με αξιοπιστία.

Το περιεχόμενο διατηρείται σε διάφορα κομβικά σημεία. Όταν ληφθεί μια αίτηση γι αυτό, αυτή δρομολογείται στο κοντινότερο ως προς τον τελικό χρήστη κομβικό σημείο, το οποίο αποστέλει τα δεδομένα με την υψηλότερη δυνατή απόδοση.

Ο προγραμματιστής αποθηκεύει τα δεδομένα σε ένα κεντρικό σημείο και ορίζει σημεία παράδοσης. Η αντιγραφή των δεδομένων στα σημεία παράδοσης γίνεται αυτόματα από τα συστήματα της Amazon. Η χρέωση γίνεται πάντα με βάση τις παραδόσεις, ο αριθμός των οποίων είναι πάντα διαθέσιμες μέσα από το Amazon CloudWatch.

AWS CloudFormation

Το AWS CloudFormation επιτρέπει στους προγραμματιστές και τους διαχειριστές ένα εύκολο μέσο να ορίζουν συλλογές από πόρους του AWS και να τους εφοδιάζουν με προκαθορισμένο και προβλέψιμο τρόπο.

Ουσιαστικά, ο χρήστης μπορεί να δημιουργήσει ένα σύνολο εφαρμογών και δυνατοτήτων μέσα από έτοιμα πρότυπα - ή να φτιάξει τα δικά του πρότυπα - και να εφοδιάζει επιλεγμένα instances με ότι ορίζεται στο πρότυπο. Για παράδειγμα, το πρότυπο μπορεί να ορίζει συγκεκριμένες εφαρμογές. Αυτές θα εγκατασταθούν με την εκκίνηση του instance. Έτσι, διατηρώντας έναν μικρό αριθμό από images και κάποια templates είναι εφικτό να έχουμε αμέτρητους συνδυασμούς με βάση τις ανάγκες μας.

Amazon Elastic MapReduce

Το MapReduce είναι μια τεχνολογία που εισήχθη το 2004 από τη Google για την κατανεμημένη επεξεργασία μεγάλων όγκων δεδομένων σε clusters. Η εν λόγω υπηρεσία είναι η υλοποίηση της Amazon και τρέχει πάνω στα Amazon Simple Storage Service μέσω του Hadoop.

Με τη χρήση αυτής της υπηρεσίας μπορεί κάποιος να προχωρήσει σε παράλληλη επεξεργασία μεγάλων όγκων δεδομένων, όπως ανάλυση αρχείων καταγραφής, data warehousing, οικονομικά δεδομένα, βιομετρικές αναλύσεις κλπ.

Δεδομένου πως τα κατανεμημένα συστήματα γίνονται όλο και πιο δημοφιλή αλλά και πιο απαραίτητα, η υπηρεσία αυτή θεωρείται από τις σημαντικότερες.

Amazon Relational Database Service (RDS)

Πρόκειται για μια υπηρεσία η οποία διευκολύνει τη ρύθμιση, εγκατάσταση, διαχείριση και κλιμάκωση μιας βάσης δεδομένων στο cloud. Προσφέρει χαμηλό κόστος, αυξομειούμενη χωρητικότητα αφαιρώντας τα ζητήματα διαχείρισης και συντήρησης.

Το Amazon RDS παρέχει πρόσβαση σε οικεία δημοφιλή συστήματα βάσεων δεδομένων, πιο συγκεκριμένα σε MySQL και Oracle. Αυτό σημαίνει πως οι υφιστάμενες εφαρμογές που χρησιμοποιούν αυτά τα συστήματα δεν χρειάζονται αλλαγές. Επιπλέον, η εταιρία δεν απεμπολεί την επένδυση που πιθανώς είχε κάνει για αγορά εργαλείων για τις συγκεκριμένες βάσεις δεδομένων. Αντίθετα, το Amazon RDS διαχειρίζεται μόνο του το patching των συστημάτων και τα backups. Μέσω του API είναι εύκολο το scaling των βάσεων τόσο σε επίπεδο υπολογιστικών πόρων όσο και σε χωρητικότητα. Με την υπηρεσία αυτή αντιμετωπίζεται το πρόβλημα της προσωρινής αποθήκευσης των δεδομένων στο instance, ενώ ταυτόχρονα αξιοποιεί γνωστά συστήματα προστατεύοντας την υπάρχουσα τεχνογνωσία και τις επενδύσεις.

Amazon ElasticCache

Η υπηρεσία Amazon ElasticCache προσφέρει τη δυνατότητα ενεργοποίησης, διαχείρισης και κλιμάκωσης ενός συστήματος memory caching στο cloud. Αυτό προσφέρει ισχυρή βελτίωση στην ταχύτητα των εφαρμογών μας αφού πολλά δεδομένα “ερχονται” από την ταχύτερη RAM αντί των αργών λειτουργιών που απαιτούν την πρόσβαση σε σκληρό δίσκο. Το Amazon ElasticCache είναι συμβατό με το πρωτόκολλο Memcached, ένας ευρέως χρησιμοποιούμενο πρωτόκολλο για memory cache εφαρμογές. Επομένως, εφαρμογές που χρησιμοποιούν μηχανισμούς cachings συμβατούς με το Memcached θα συνεργαστούν άμεσα και χωρίς περαιτέρω εργασία με το Amazon ElasticCache.

Amazon Simple Notification Service

Πρόκειται για μια υπηρεσία ειδοποιήσεων μέσω θεματικών περιοχών (topics). Εκεί η εφαρμογή ή ο χρήστης κοινοποιούν ενημερώσεις οι οποίες είναι διαθέσιμες σε εγγεγραμμένες εφαρμογές ή χρήστες. Πρακτικά είναι μια cloud υλοποίηση των μηχανισμών topics των message brokers.

Μέσα από ένα απλό interface ο χρήστης ή η εφαρμογή μπορούν να δημιουργήσουν topics προκειμένου να ενημερώνουν άλλους χρήστες ή εφαρμογές. Οι τελευταίοι οφείλουν μέσω του interface να εγγράφονται σε αυτά τα topics. Έπειτα, οποιαδήποτε ενημέρωση θα τους κοινοποιείται με τη χρήση του επιλεγμένου πρωτοκόλλου, όπως http, email κ.α.

Ανάμεσα στις πιθανές χρήσεις αυτής της υπηρεσίας είναι η παρακολούθηση των εφαρμογών, work-flow συστήματα, ενημέρωση time-sensitive εφαρμογών, κινητά κ.α.

Amazon Simple Queue Service (SQS)

Πρόκειται για την cloud υλοποίηση των μηχανισμών queue των message brokers. Η διαφορά με την προαναφερθείσα υπηρεσία έχει να κάνει με το ότι τα μηνύματα σε μια ουρά αποθηκεύονται μέχρι να καταναλωθούν, ενώ η εφαρμογή που τα λαμβάνει είναι μία. Αφ’ ενός λοιπόν πρόκειται για τη μεταφορά μηνυμάτων που δεν έχουν προσωρινή αξία, πχ η τιμή μιας μετοχής, η οποία αλλάζει ανά μερικά λεπτά και αφ’ εταίρου για μηνύματα τα οποία προορίζονται για επεξεργασία από συγκεκριμένες εφαρμογές και όχι για γενική ενημέρωση πχ η επεξεργασία μιας νέας παραγγελίας, η οποία γίνεται ασύγχρονα ώστε να μην μπλοκάρεται το User Interface αλλά δεν έχει νόημα να γίνεται πολλές φορές.

AWS Identity and Access Management (IAM)

Το Amazon Web Services Identity and Access Management είναι μια υπηρεσία η οποία προσφέρει την ασφαλή αδειοδότηση χρηστών στις υπηρεσίες του cloud. Επιτρέπει τη

δημιουργία και τη διαχείριση χρηστών στο AWS και τον ορισμό δικαιωμάτων χρηστών που τους διαχειρίζεται το εταιρικό directory. Έτσι, είναι εφικτό το federation χρηστών του εταιρικού καταλόγου και των υπηρεσιών της Amazon Web Services. Αυτό σημαίνει πως η διαχείριση δικαιωμάτων σε υπηρεσίες όπως τα S3 buckets μπορεί να γίνεται από το εταιρικό directory (πχ Active Directory, OpenLDAP) χωρίς να απαιτείται η διπλή συντήρηση των δεδομένων των χρηστών σε αυτό και στην Amazon.

Εργαλεία ανάπτυξης για το Amazon's EC2

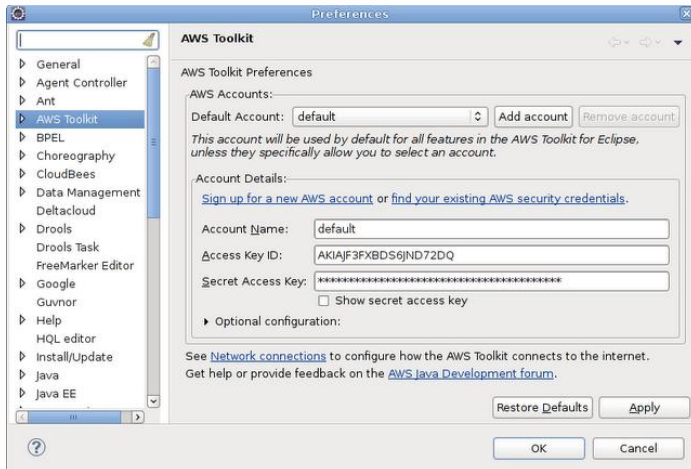
Η Amazon προσφέρει ένα πρόσθετο για το περιβάλλον εργασίας eclipse (www.eclipse.com), το οποίο μπορούμε να εγκαταστήσουμε από τη διεύθυνση <http://aws.amazon.com/eclipse/> . Συνιστάται να έχουμε το eclipse for java enterprise development, σε έκδοση 3.6 και άνω, δηλαδή helios ή indigo. Αυτό θα μας εξασφαλίσει ότι έχουμε όλα τα προαπαιτούμενα για το πρόσθετο του amazon.

Μαζί με το πρόσθετο αυτό, θα εγκατασταθεί και το Java SDK, ένα σύνολο βιβλιοθηκών που προσφέρει η Amazon ώστε να μπορούμε να διαχειριστούμε τους πόρους και τις υπηρεσίες που μας παρέχονται προγραμματιστικά. Για παράδειγμα, μπορούμε μέσω εντολών του SDK να δημιουργήσουμε νέα Queues στο SQS, να ρυθμίσουμε τις καταγραφές της εφαρμογής μας, να προσπελάσουμε βάσεις δεδομένων στο RDS και άλλα.

Το πρόσθετο αυτό μας επιτρέπει επίσης να διαχειριστούμε τους πόρους που έχουμε ενοικιάσει μέσα από το ίδιο το eclipse καθώς και να εγκαταστήσουμε αλλά και να αποσφαλματώσουμε (debugging) εφαρμογές στο Beanstalk. Τέλος, μπορούμε να ανεβάσουμε EC2 instances.

Εκτός από τη Java, υπάρχουν SDK και για πλήθος άλλων γλωσσών όπως η PHP, η Ruby, η πλατφόρμα [.net](http://net), το Android, το iOS.

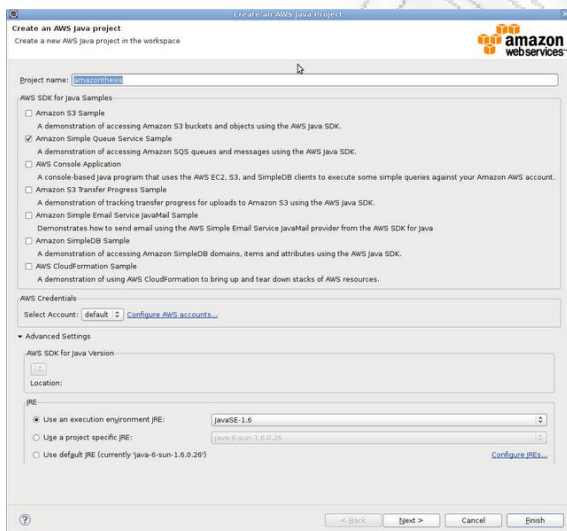
Αφού το εγκαταστήσουμε θα πρέπει να επανεκκινήσουμε το eclipse, και να πάμε Windows->preferences->AWS Toolkit να δηλώσουμε τα στοιχεία μας.



Τα στοιχεία μας μπορούμε να τα βρούμε στη σελίδα Account στο Amazon console.

Επίσης, θα πρέπει να έχουμε εγγραφεί σε όλες τις υπηρεσίες που θέλουμε να χρησιμοποιήσουμε. Αφού έχουμε ολοκληρώσει επιτυχώς τα προηγούμενα βήματα, μπορούμε να δημιουργήσουμε ένα νέο AWS Java Project. Συμπληρώνουμε το επιθυμητό όνομα και επιλέγουμε τις υπηρεσίες που θέλουμε. Το πρόγραμμα που θα δημιουργηθεί θα είναι μια standalone εφαρμογή η οποία θα χρησιμοποιεί πόρους στο Amazon.

Μπορούμε να δημιουργήσουμε διάφορα projects ώστε να δούμε όλες τις λειτουργίες που παρέχονται από το SDK. Στο παράδειγμά μας δημιουργούμε ένα νέο έργο για το SQS.



Στον κώδικα που παραθέτουμε βλέπουμε πώς με απλές εντολές του SDK μπορούμε να δημιουργήσουμε μία Queue στο SQS και να στείλουμε ένα μήνυμα.

[Java excerpt]

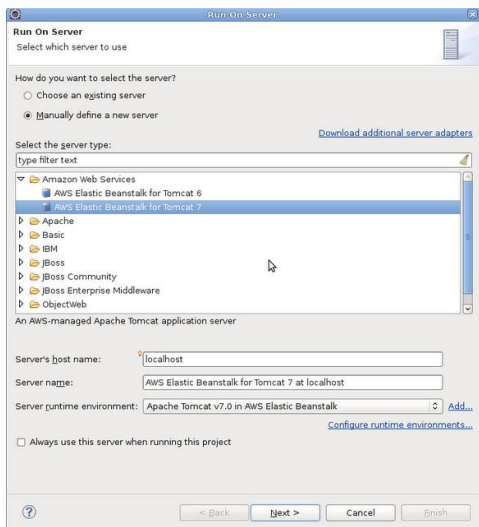
```
AmazonSQS sqs = new AmazonSQSClient(new PropertiesCredentials(
    SimpleQueueServiceSample.class.getResourceAsStream(
        "AwsCredentials.properties")
    )
);
try {
    //create a queue
    CreateQueueRequest createQueueRequest = new
CreateQueueRequest("MyQueue");
    String myQueueUrl =
sqs.createQueue(createQueueRequest).getQueueUrl();

    // Send a message
    sqs.sendMessage(
        new SendMessageRequest(myQueueUrl, "This is my message
text.")
    );
} catch (AmazonServiceException ase) {
    ...
}
```

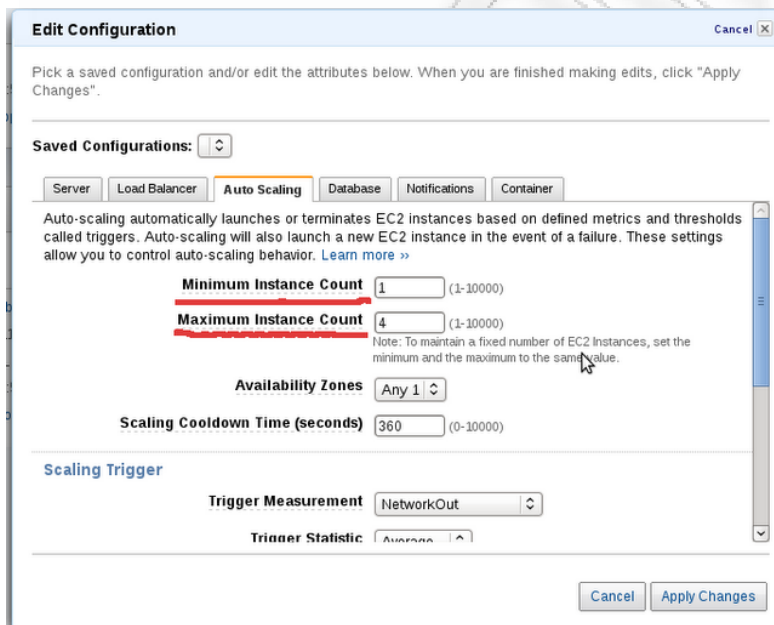
Εάν θέλουμε να δημιουργήσουμε μια εφαρμογή που θα εκτελείται στο Beanstalk, πρέπει να δημιουργήσουμε ένα project τύπου "AWS Java Web Project". Συμπληρώνουμε το όνομα και πατάμε Finish.

Έπειτα, κάνουμε δεξί κλικ στο έργο, και επιλέγουμε Amazon Web Services->Deploy to AWS Elastic Beanstalk... Θα εμφανιστεί η οθόνη deployment του eclipse. Εκεί θα πρέπει να επιλέξουμε ως server, το "AWS Elastic Beanstalk for Apache Tomcat 7" και στα επόμενα βήματα να αφήσουμε τις προεπιλεγμένες τιμές. Πρέπει να αναφέρουμε ότι καλό θα είναι βέβαια, σε συνθήκες πραγματικής ανάπτυξης να ορίσουμε την ετικέτα της έκδοσης (version label) την οποία ανεβάζουμε γιατί όπως θα δούμε μπορεί να χρησιμοποιηθεί από την κονσόλα της Amazon ώστε να εκκινήσουμε μια διαφορετική έκδοση της εφαρμογής μας.

Μετά από κάποιο χρονικό διάστημα η εφαρμογή μας θα είναι εγκατεστημένη στο Elastic Beanstalk.



Μετά την ολοκλήρωση της διαδικασίας μπορούμε να δούμε και να ρυθμίσουμε τις παραμέτρους της εφαρμογής μας στο amazon console. Αξίζει να παρουσιάσουμε τα χαρακτηριστικά του Auto-scaling.



Στην παραπάνω οθόνη βλέπουμε τις παραμέτρους για το auto scaling, ο ελάχιστος και ο μέγιστος αριθμός instances για την εφαρμογή μας. Στην επόμενη οθόνη, βλέπουμε τα κριτήρια με τα οποία θα πραγματοποιείται το auto-scaling.

Edit Configuration Cancel

Pick a saved configuration and/or edit the attributes below. When you are finished making edits, click "Apply Changes".

Saved Configurations:

Server Load Balancer **Auto Scaling** Database Notifications Container

Scaling Trigger

Trigger Measurement: NetworkOut

Trigger Statistic: Average

Unit of Measurement: Bytes

Measurement Period (minutes): 5 (1-600)

Breach Duration (minutes): 5 (1-600)

Upper Threshold: 6000000 (0-20000000)

Upper Breach Scale Increment: 1

Lower Threshold: 2000000 (0-20000000)

Lower Breach Scale Increment: -1

Cancel Apply Changes

Από τις άλλες καρτέλες, μπορούμε εάν επιθυμούμε να αλλάξουμε τις ρυθμίσεις της εφαρμογής μας, τα χαρακτηριστικά του server, του load balancer, τη βάση που θα χρησιμοποιεί, τις ειδοποιήσεις και τον container.

Σημαντικό χαρακτηριστικό, ειδικά για τον κύκλο ανάπτυξης αλλά και τον κύκλο ζωής του λογισμικού είναι η δυνατότητα εκκίνησης της εφαρμογής σε μια διαφορετική έκδοση.

Deploy a Different Version Cancel

Select a different version to deploy to this environment, or upload a new version and deploy it to this environment.

Environment: thesisweb

Deployment

Deploy an existing version

Upload and deploy a new version

Version Label:

Description: (optional, 200 char maximum)

Upload Existing Application: (e.g. WAR file)

Αναζήτηση...

Cancel Deploy Version

Σαν τελευταία εικόνα αξίζει να παρουσιαστούν οι δυνατότητες διαχείρισης που μας παρέχει το εν λόγω πρόσθετο μέσα από το περιβάλλον ανάπτυξης.

Image ID	Root Device Type	State	Type	Availability Zone	Key Pair	Launch Time	Security Groups
ami-8dde1f04	ebs	stopping	t1.micro	us-east-1a		5 Σεπ 2011 12:53:39 πμ	elasticbeanstalk-default

Μέσα από αυτό μπορούμε να διαχειριστούμε τα instances, τα images (AMI), τον αποθηκευτικό χώρο και κάποιες ρυθμίσεις ασφαλείας.

Eucalyptus

Το eucalyptus είναι ακρωνύμιο του “Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems” και είναι λογισμικό για την υλοποίηση ιδιωτικών clouds σε computer clusters. Είναι διαθέσιμο τόσο για εταιρίες (open-core version) καθώς και ως

λογισμικό ανοιχτού κώδικα (open-source). Διαθέτει μια διεπαφή η οποία είναι συμβατή με το Amazons EC2 και το S3 αλλά είναι δομημένο με τέτοιο τρόπο που να υποστηρίζει ταυτόχρονα και άλλες διεπαφές (interfaces). Μπορεί να τρέξει πάνω στις περισσότερες διανομές Linux και υποστηρίζει αρκετές τεχνολογίες virtualization (VMWare, Xen και KVM supervisors).

Το eucalyptus ξεκίνησε ως ερευνητική εργασία στο χώρο του high performance computing (HPC) στο τμήμα πληροφορικής του Πανεπιστημίου της Santa Barbara, στην California, ενώ το 2009 ιδρύθηκε η εταιρία Eucalyptus Systems Inc με σκοπό την εμπορικοποίηση της πλατφόρμας.

Μέχρι πρόσφατα το eucalyptus ήταν η βασική πλατφόρμα cloud για τη διανομή Ubuntu, η οποία όμως αποφάσισε την αντικατάστασή του με το OpenStack από την έκδοση 11.10. Αυτό βέβαια δεν σημαίνει πως δεν θα υποστηρίζεται - αντίθετα, έχει ήδη μπει στα repositories της Debian - απλώς το τελευταίο θα είναι η προκαθορισμένη επιλογή.

Με το eucalyptus είναι εφική η δημιουργία ιδιωτικών εσωτερικών και υβριδικών clouds και παρέχει λειτουργίες για τη χρήση πόρων του εσωτερικού cloud όσο και πόρων δημοσίου cloud - στην περίπτωση υλοποίηση υβριδικού cloud. Ο σχεδιασμός του είναι τμηματικός (modular) και επεκτάσιμος (extensible) με μια Web-Services based αρχιτεκτονική που επιτρέπει στο eucalyptus να εκθέτει μια ποικιλία από APIs. Επίσης, αν και παρέχει το δικό του σύνολο εργαλείων, με το όνομα eucatools, για τη διαχείριση των πόρων του, υλοποιεί ένα API συμβατό με αυτό της Amazon κάτι που επιτρέπει τη διασυνεργασία αυτού με υπάρχοντα εργαλεία και υπηρεσίες του Amazons EC2.

Τα βασικά χαρακτηριστικά που προσφέρει το eucalyptus είναι τα εξής:

- Συμβατότητα με το API του Amazon EC2
- Εγκατάσταση από native πακέτα των βασικότερων διανομών Linux (deb και rpm packages)
- Ασφαλή ενδοεπικοινωνία με χρήση ασφαλών SOAP και Web Services.
- Υποστήριξη τόσο για Linux όσο και Windows Virtual Machines
- Elastic IPs και ομάδες ασφαλείας
- Διαχείριση χρηστών και ομάδων

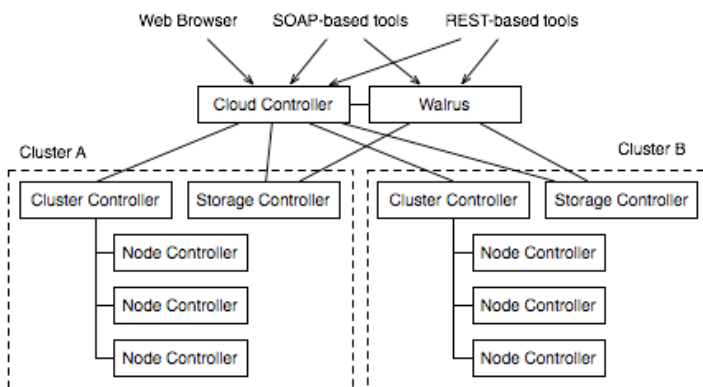
Αρχιτεκτονική

Η αρχιτεκτονική του eucalyptus εν συντομία περιλαμβάνει πέντε βασικά συστατικά υψηλού επιπέδου: Cloud Controller (CLC), Cluster Controller (CC), Walrus, Storage Controller (SC) και Node Controller (NC).

Κάθε συστατικό από αυτά διαθέτει το δικό web interface και είναι υλοποιημένο ως αυτόνομο web service. Αυτό παρέχει δύο σημαντικά πλεονεκτήματα. Το πρώτο είναι ότι κάθε συστατικό παρέχει τη λειτουργικότητά του μέσω από ένα API ανεξάρτητο από γλώσσα προγραμματισμού, αφού παρέχεται με τη μορφή WSDL που περιγράφει τόσο τις λειτουργίες του όσο και τις δομές

δεδομένων που υποστηρίζει και καταλαβαίνει. Το δεύτερο πλεονέκτημα είναι ότι αξιοποιούνται γνωστές και διαδεδομένες τεχνολογίες ασφαλείας των Web services, για την ασφαλή επικοινωνία ανάμεσα στα επιμέρους συστατικά.

Η αρχιτεκτονική του φαίνεται στο επόμενο σχήμα.



Συστατικά του eucalyptus

Cloud Controller (CLC): Είναι το component που διαχειρίζεται όλο το cloud και είναι υπεύθυνο για την έκθεση (exposing) και διαχείριση των πόρων. Διαθέτει EC2 συμβατό API καθώς και web interface.

Cluster Controller (CC): Είναι το component που διαχειρίζεται ένα cluster, δηλαδή ένα κομμάτι του cloud και αποτελεί τον ενδιάμεσο ανάμεσα στον Cloud Controller και τους Node Controllers. Επίσης, δίνει αναφορά για τους πόρους του cluster στον Cloud Controller και κάνει τη διαχείριση της κίνησης στην περίπτωση που χρησιμοποιούμε virtual network στις εικονικές μηχανές.

Walrus: Είναι μια υλοποίηση αποθηκευτικού χώρου συμβατό με το S3 της Amazon, παρέχοντας έναν μηχανισμό μόνιμης αποθήκευσης στους χρήστες. Επίσης, χρησιμοποιείται για την αποθήκευση των images των λειτουργικών συστημάτων που μπορούν να τρέξουν ανά πάσα στιγμή στο cloud.

Storage Controller (SC): Παρέχει υπηρεσίες αποθηκευτικού χώρου επιπέδου block τα οποία μπορούν να προσαρτηθούν στα VMs. Με άλλα λόγια δημιουργεί και διαχειρίζεται εικονικούς δίσκους τους οποίους χρησιμοποιούν οι εικονικές μηχανές. Επίσης, μπορεί να δημιουργήσει στιγμιότυπα (snapshots) από συγκεκριμένες χρονικές στιγμές των δίσκων. Μάλιστα, ένα στιγμιότυπο μπορεί να χρησιμοποιηθεί και από μια άλλη εικονική μηχανή. Η υλοποίηση είναι συμβατή με το EBS της Amazon.

Node Controller (NC): Είναι μηχανήματα με virtualization extensions στους επεξεργαστές. Πάνω σε αυτά τρέχουν οι hypervisors (KVM ή XEN) και αρακολουθούν την εκτέλεση των Virtual Machines και είναι υπεύθυνοι για την εκκίνηση, την εκτέλεση και τον τερματισμό τους.

OpenNebula

Το OpenNebula είναι ένα εργαλείο ανοιχτού κώδικα για το cloud computing με δυνατότητες διαχείρισης ετερογενών καταμεμημένων data centers. Το OpenNebula διαχειρίζεται τις υποδομές του data center και προσφέρει τη δυνατότητα δημιουργίας εσωτερικών ιδιωτικών, δημοσίων και υβριδικών clouds παροχής υπηρεσιών υποδομής (IaaS). Διαθέτει υπηρεσίες διαχείρισης των πόρων του cloud, μηχανισμούς παρακολούθησης και ασφαλείας και μπορεί να διαχειριστεί πόρους τόσο του εσωτερικού όσο και εξωτερικών clouds, καταναίμοντάς τους σύμφωνα με προκαθορισμένες πολιτικές. Υποστηρίζεται από την Ευρωπαϊκή Ένωση και θεωρείται πως θα παίξει καθοριστικό ρόλο στο χώρο του cloud computing, αφού δίνει έμφαση στα standards, το interoperability και το portability παρέχοντας στους χρήστες και τους διαχειριστές ελευθερία επιλογών ανάμεσα σε διεπαφές που είναι συμβατές με το Amazon EC2, το OGF OCC1 και το vCloud. Επίσης η ευέλικτη αρχιτεκτονική του προσφέρει συνεργασία με διάφορους hypervisors όπως KVM και XEN και μπορεί να διαχειριστεί πολλούς συνδυασμούς hardware και software σε ένα data center.

Το project είναι εξαιρετικά ενεργό, με έντονη δραστηριότητα. Ήταν από τους καθοδηγητές στο Google Summer of Code το 2010 και υποστηρίζεται από την C12G, μια εταιρία που συνεργάζεται με το Επιστημονικό Πάρκο της Μαδρίτης.

Το OpenNebula μπορεί να συνεργαστεί με το OpenStack για τη δημιουργία clouds ενώ είναι διαθέσιμο για τις βασικότερες διανομές Linux.

Συμπεράσματα

Λαμβάνοντας υπόψιν τις υπηρεσίες που προσφέρει ένα cloud παροχής υπηρεσιών υποδομής μπορούμε να αντιληφθούμε ότι πλέον είναι εφικτή η υλοποίηση εφαρμογών που θα τρέχουν στο cloud με το μέγιστο βαθμό ελέγχου τόσο των εφαρμογών όσο και των πόρων που απαιτούνται. Οι υπηρεσίες στο επίπεδο υποδομής εκτός από αποθηκευτικό χώρο και επεξεργαστική ισχύ έχουν εμπλουτιστεί με τους περισσότερους από τους βασικούς μηχανισμούς που χρησιμοποιούνται στα σύγχρονα υπολογιστικά συστήματα όπως σχεσιακές βάσεις δεδομένων, συστήματα ασύγχρονης ανταλλαγής μηνυμάτων (message brokers), memory caching και load balancing. Επιπλέον, προστίθενται και νέοι μηχανισμοί που κερδίζουν συνεχώς σε δημοτικότητα και συνδέονται περισσότερο με το cloud όπως noSQL καταμεμημένες βάσεις.

Η σαφή υπεροχή της πρωτοπόρου Amazon, η οποία επέσπευσε την εξέλιξη του cloud computing, την έχει καταστήσει de facto standard. Έτσι, οι νέες λύσεις που προτείνονται όπως

το eucalyptus τείνουν να είναι συμβατές με το Amazon's EC2 μειώνοντας τον κίνδυνο του vendor lock-in, αυξάνοντας τον ανταγωνισμό αλλά και προσφέροντας λύσεις για ιδιωτικά εσωτερικά clouds, ανοίγοντας το δρόμο για την ευρύτερη αποδοχή του cloud computing ακόμα και από επιχειρήσεις οι οποίες αντιμετωπίζουν νομικά ή και άλλα θέματα στη χρήση των δημοσίων clouds.

Platform as a Service

Στην ενότητα αυτή θα αναφερθούμε σε παρόχους υπηρεσιών πλατφόρμας. Οι υπηρεσίες αυτές βρίσκονται σε υψηλότερο επίπεδο από αυτό της υποδομής, με την έννοια ότι προσφέρουν έτοιμες λύσεις αφαιρώντας ταυτόχρονα τη διαχείριση ζητημάτων κατώτερου επιπέδου από το χρήστη. Έτσι, για παράδειγμα, ενώ στις υπηρεσίες υποδομής παρέχεται ένα virtual machine στο οποίο πρέπει να εγκαταστήσουμε το απαιτούμενο λογισμικό για την εκτέλεση της εφαρμογής μας και να μεριμνήσουμε για την μόνιμη αποθήκευση των δεδομένων εμπλέκοντας με τις υπηρεσίες αποθηκευτικού χώρου, στο επίπεδο της πλατφόρμας δεν οφείλουμε και δεν μπορούμε να ασχοληθούμε με αυτά. Τα instances, το λειτουργικό τους σύστημα και ο αποθηκευτικός χώρος είναι ήδη καθορισμένα από τον πάροχο. Σε εμάς απομένει το deployment της εφαρμογής και ο έλεγχος των πόρων που απαιτούνται σε υψηλότερο επίπεδο, χωρίς να έχουμε ορατότητα και έλεγχο στο πώς καλύπτονται.

Όπως και στο προηγούμενο κεφάλαιο, αφού παρουσιάσουμε τις βασικές υπηρεσίες που πρέπει να προσφέρονται στο επίπεδο υποδομής, θα αναφερθούμε στους κυριότερους παρόχους.

Βασικές υπηρεσίες υποδομής

Το συγκεκριμένο επίπεδο είναι το πλέον ασαφές όσον αφορά τις υπηρεσίες που πρέπει να προσφέρονται. Εκτός των κοινών υπηρεσιών που οφείλει να παρέχει κάθε πάροχος υπηρεσιών cloud, είναι αυτονόητο πως θα πρέπει να προσφέρεται ένα περιβάλλον εκτέλεσης (runtime) των εφαρμογών των χρηστών ανάλογα με το είδος της πλατφόρμας. Επομένως, μια πλατφόρμα java θα πρέπει να προσφέρει τουλάχιστον έναν Java Application Server. Αντίστοιχα ένα περιβάλλον PHP θα πρέπει να προσφέρει έναν Web Server με PHP κ.ο.κ. Συν τοις άλλοις, τα δεδομένα του χώρου ορίζουν και την παροχή βασικών υπηρεσιών, όπως βάσεις δεδομένων, συστήματα ασύγχρονης επικοινωνίας κ.α. Πολλά από αυτά, λόγω της σημασίας τους, προσφέρονται ήδη στο επίπεδο των υπηρεσιών υποδομής - οπότε είναι αυτονόητο ότι θα πρέπει να προσφέρονται και σε αυτό το επίπεδο.

Παρακάτω θα παρουσιάσουμε τους κυριότερους παρόχους υπηρεσιών υποδομής, με μία σύντομη αναφορά στις υπηρεσίες που παρέχει ο καθένας, εξετάζοντας τις υπηρεσίες αυτές υπό το πρίσμα των απαιτήσεων του cloud computing.

Google App Engine

Το Google App Engine παρέχει τη δυνατότητα να εκτελούμε τις εφαρμογές μας στα συστήματα της Google. Κάνει εύκολη την εγκατάσταση της εφαρμογής, τη διαχείρισή της και την κλιμάκωσή της χωρίς να χρειάζεται η συντήρηση κάποιου server. Η διαφορά από τις υπηρεσίες υποδομής είναι ότι ο χρήστης δεν έχει έλεγχο στα χαρακτηριστικά του server, πχ μνήμη, επεξεργαστική ισχύ κλπ. Επιπλέον, δεν είναι υποχρεωμένος να διατηρεί ενημερωμένο το image αφού δεν υπάρχει πλέον κάτι τέτοιο. Το Google App Engine παρέχει και ελέγχει τον application server στον οποίο θα εκτελεστεί η εφαρμογή.

Ένα άλλο αξιόλογο χαρακτηριστικό του Google App Engine είναι η ευκολία διασύνδεσης των εφαρμογών που αναπτύσσονται για το Google App Engine με τις εφαρμογές που ήδη παρέχει το Google υπό το πρίσμα του Software as a Service.

Οι κυριότερες υπηρεσίες που προσφέρει η πλατφόρμα αναλύονται παρακάτω.

Υπηρεσίες του Google App Engine

Runtime environments

Το Google App Engine παρέχει application servers για εκτέλεση εφαρμογών σε Java, Python και Go. Η τελευταία είναι μια νέα γλώσσα για κατανεμημένο προγραμματισμό. Περισσότερες πληροφορίες διατίθενται στο <http://golang.org/>.

Go Runtime Environment

Το περιβάλλον αυτό επιτρέπει την εκτέλεση εφαρμογών γραμμένων στη γλώσσα Go. Το αντίστοιχο SDK περιλαμβάνει ένα αυτόματο μηχανισμό build. Επίσης, ο χρήστης δεν χρειάζεται να κάνει build και να ανεβάσει το εκτελέσιμο. Αντίθετα, το build πραγματοποιείται αυτόματα όταν ανιχνευθούν αλλαγές στον πηγαίο κώδικα.

Μια άλλη παρατήρηση είναι ότι τα APIs που παρέχει το Google App Engine για τη γλώσσα αυτή είναι ελαφρώς διαφοροποιημένα - κάτι αρκετά σημαντικό για τις υφιστάμενες εφαρμογές αλλά και για ενδεχόμενο vendor lock-in.

Python Runtime Environment

Με το συγκεκριμένο περιβάλλον εκτέλεσης είναι εφικτή η εκτέλεση εφαρμογών στην αντίστοιχη γλώσσα στο Google App Engine. Χρησιμοποιείται ένας βελτιστοποιημένος interpreter. Υπάρχουν διαθέσιμες οι standard βιβλιοθήκες της Python, βιβλιοθήκες για τη χρήση των υπηρεσιών που παρέχει το Google App Engine ενώ ο χρήστης μπορεί να ανεβάσει και να χρησιμοποιήσει επιπλέον βιβλιοθήκες, δικές του ή τρίτων.

Java Runtime Environment

Το περιβάλλον εκτέλεσης Java προσφέρει τη δυνατότητα εκτέλεσης εφαρμογών που μπορούν να εκτελεστούν σε μία Java Virtual Machine. Αυτό σημαίνει όχι μόνο Java εφαρμογές αλλά και Scala, JRuby, Groovy και άλλες.

Το περιβάλλον είναι συμβατό με Java 5 και Java 6. Η χρήση των υπηρεσιών του Google App Engine γίνεται μέσω του προσφερόμενου SDK ενώ η επικοινωνία της εφαρμογής με το περιβάλλον μέσω των Java Servlets Standards.

Datastore

Το Google App Engine παρέχει μια υπηρεσία καταμεμημένης αποθήκευσης δεδομένων με αυτόματη αυξομείωση του παρεχόμενου χώρου βάσει των αναγκών της εφαρμογής. Αν και δεν ακολουθεί τις παραδοσιακές σχεσιακές βάσεις και ανήκει στο χώρο των schemeless ή soft-schema βάσεων, τα JDO/JPA Java interfaces που παρέχονται καθώς και το Python interface επιτρέπουν τη λειτουργία με δομημένα δεδομένα όπως στις σχεσιακές βάσεις.

Αξίζει να τονιστεί πως ο χρήστης μπορεί να επιλέξει ανάμεσα σε δύο διαφορετικούς μηχανισμούς datastore που διαφοροποιούνται ως προς τις εγγυήσεις διαθεσιμότητας και συνέπειας των δεδομένων τους.

Google Accounts

Το Google App Engine προσφέρει ενοποίηση (integration) με τους λογαριασμούς χρηστών του Google για την πιστοποίηση των χρηστών. Έτσι, οι τελικοί χρήστες της εφαρμογής μπορούν να συνδέονται - εφ' όσον έχουν το δικαίωμα - χωρίς να εγγράφονται και να δημιουργούν ξεχωριστό λογαριασμό στην εφαρμογή.

Αξιοσημείωτο είναι μάλιστα πως μέσα από το User Accounts API είναι εφικός ο έλεγχος εξουσιοδότησης. Για παράδειγμα, μέσα από το API μπορεί η εφαρμογή να λάβει πληροφορίες για το αν ο χρήστης είναι καταχωρημένος ως διαχειριστής της εφαρμογής και φυσικά να εμφανίσει επιπλέον επιλογές.

Caching

Πρόκειται για μια υπηρεσία memory caching με στόχο την επιτάχυνση εκτέλεσης των εφαρμογών.

Scheduled Tasks

Πρόκειται για υπηρεσία προγραμματισμού εργασιών. Μέσα από αυτήν είναι εφικτός ο χρονικός προγραμματισμός εργασιών που πρέπει να εκτελεστούν κάποια δεδομένη χρονική στιγμή ή επαναλαμβανόμενα. Για παράδειγμα υπολογισμός χρεώσεων κάθε τέλος του μήνα.

Task Queues

Η υπηρεσία αυτή παρέχει ουρές στις οποίες μπορούν να καταχωρούνται εργασίες για ασύγχρονη επεξεργασία.

Integrated services

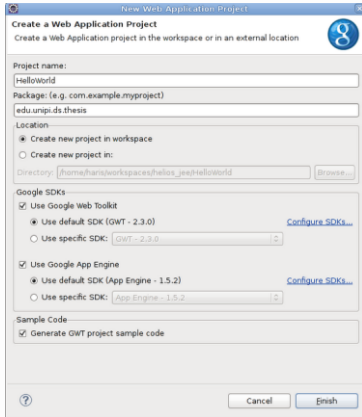
Με τον όρο αυτό νοείται η ενοποίηση υπηρεσιών που ήδη παρέχει το google σε κάποια άλλη μορφή με την εφαρμογή του χρήστη. Για παράδειγμα, η αποστολή mail, η διαχείριση εικόνων, η συνεργασία με άλλες εφαρμογές Google όπως google calendar κ.α.

Ανάπτυξη για το Google App Engine

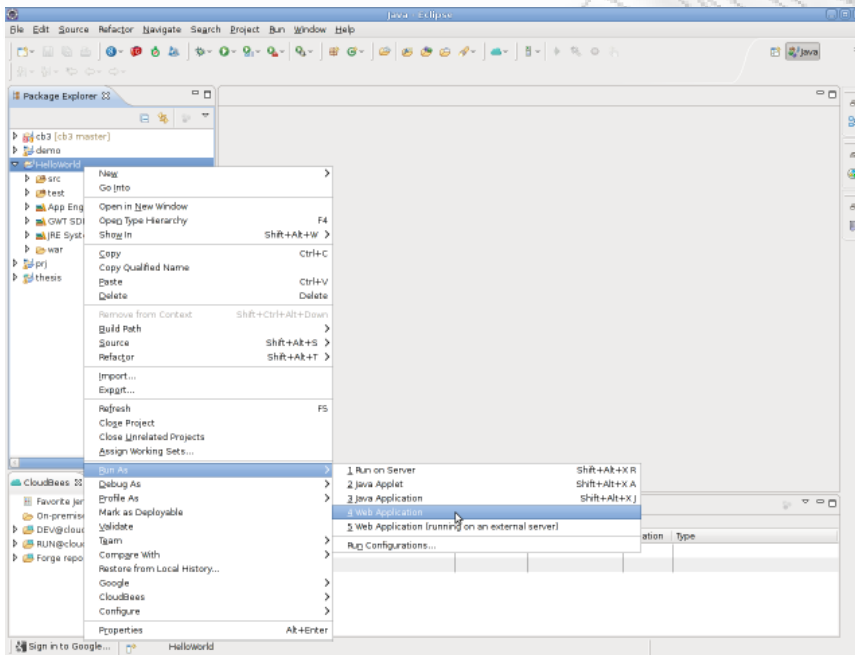
Η Google παρέχει ένα πρόσθετο για το περιβάλλον ανάπτυξης eclipse (<http://www.eclipse.com>). Με το πρόσθετο αυτό μπορούμε να αναπτύξουμε τοπικά την εφαρμογή μας, να την εκτελέσουμε τοπικά σε ένα περιβάλλον που προσομοιώνει το περιβάλλον εκτέλεσης του Google App Engine και να την κάνουμε deploy στην εν λόγω πλατφόρμα.

Το πρόσθετο είναι διαθέσιμο από τις σελίδες του Google App Engine ανάλογα με την έκδοση του eclipse. Για την έκδοση helios (3.6) η διεύθυνση είναι <http://dl.google.com/eclipse/plugin/3.6>

Αφού εγκαταστήσουμε το πρόσθετο, επανεκκινούμε το eclipse και πάμε να δημιουργήσουμε ένα νέο Web Application Project. Συμπληρώνουμε τα στοιχεία για την εφαρμογή και πατάμε Finish.



Έπειτα, “τρέχουμε” την εφαρμογή κάνοντας επάνω της δεξιά κλικ->Run as->Web application



Στο παράθυρο που θα ανοίξει μπορούμε να δοκιμάσουμε την εφαρμογή μας. Εάν όλα είναι σωστά μπορούμε να την ανεβάσουμε στο Google App Engine. Για να ολοκληρώσουμε τη διαδικασία θα πρέπει να προσθέσουμε μια εφαρμογή στο <https://appengine.google.com/start/createapp> με το επιθυμητό Application ID.

Μόλις ολοκληρωθεί η διαδικασία επιτυχώς, η σελίδα που θα εμφανιστεί θα μας ενημερώσει για την επιτυχία και θα μας προτρέψει να προσθέσουμε administrators και χρήστες στην εφαρμογή.

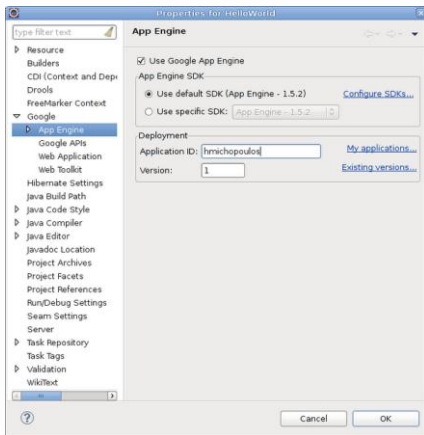
The screenshot shows the Google App Engine console after successful application registration. The page title is "Application Registered Successfully". It provides information about the application identifier (hmicopoulos), the storage scheme (High Replication), and Google authentication. A list of options is provided: "View the dashboard for HelloWorld", "Use appcfg to upload and deploy your application code", and "Add administrators to collaborate on this application". The "Add administrators" option is highlighted with a red underline.

Στη σελίδα αυτή, εκτός από το User Account, μπορούμε να δούμε πληροφορίες για όλες τις υπηρεσίες του Google App Engine σχετικά με την εφαρμογή μας.

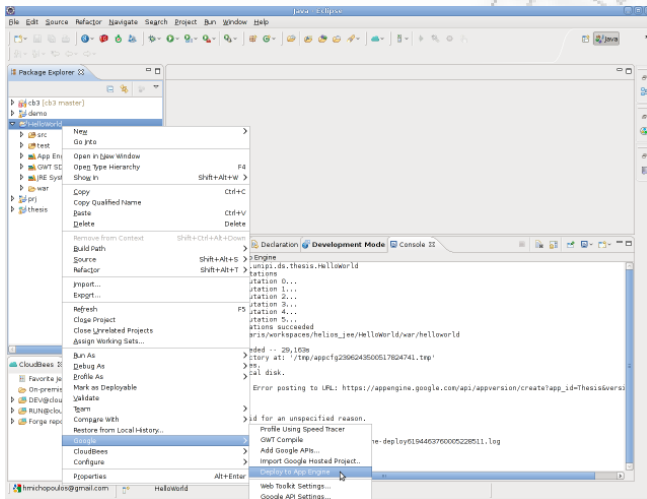
The screenshot shows the "Permissions" page in the Google App Engine console. It displays a table of users with columns for "Google Account", "Role", "Status", and "Remove Access". The current user, hmicopoulos@gmail.com, is listed as the "Owner" with an "Active" status. Below the table, there is a section titled "Invite a user to collaborate on this application" with an "Email" input field and a "Role" dropdown menu set to "Developer". A left sidebar contains navigation links for various services like Dashboard, Instances, Logs, Versions, etc.

Google Account	Role	Status	Remove Access
hmicopoulos@gmail.com — you	Owner	Active	Remove

Έπειτα, πρέπει στο eclipse να δηλώσουμε αυτό το Application ID κάνοντας στην εφαρμογή μας δεξί κλικ->Properties->Google->App Engine.



Αφού ολοκληρώσουμε τα βήματα αυτά επιτυχώς μπορούμε να ανεβάσουμε την εφαρμογή μας στο Google App Engine κάνοντας δεξί κλικ στην εφαρμογή->Google->Deploy to App Engine



Η εφαρμογή μας θα είναι διαθέσιμη στη διεύθυνση <http://applicationId.appspot.com>, στο παράδειγμα <http://hmichopoulos.appspot.com>.

AppScale

Πρόκειται για μια πλατφόρμα υβρικών clouds. Το AppScale υλοποιεί έναν σημαντικό αριθμό από δημοφιλή APIs μεταξύ αυτών αυτό του Google App Engine, του MapReduce (μέσω του Hadoop), του MPI και άλλα. Εκτελείται σαν guest εικονική μηχανή πάνω σε ένα virtualization layer το οποίο μπορεί να φιλοξενήσει images του Ubuntu Karmic Coala. Από πλευράς hypervisors υποστηρίζει το Xen και το KVM. Μπορεί να εγκατασταθεί τόσο στο Amazon's EC2 όσο και σε ιδιωτικά εσωτερικά clouds που τρέχουν σε eucalyptus. Υποστηρίζει, όπως και το Google App Engine, εφαρμογές γραμμένες σε Python, Java και Go.

CloudBees

Η πλατφόρμα αυτή μπορεί να θεωρηθεί περισσότερο ως πλατφόρμα ανάπτυξης και λιγότερο ως πλατφόρμα εκτέλεσης εφαρμογών, χωρίς αυτό να υποβαθμίζει τον δεύτερο ρόλο της. Διακρίνεται για την πληθώρα των εργαλείων που προσφέρει όσον αφορά τον κύκλο ανάπτυξης παρέχοντας σημαντικά εργαλεία ενώ όπως ήδη είδαμε έρχεται μαζί με ένα πρόσθετο (plugin) για ένα παραδοσιακό περιβάλλον ανάπτυξης, το eclipse. Ταυτόχρονα, μέσω APIs επικοινωνεί και με διαδικτυακά περιβάλλοντα ανάπτυξης, όπως το cloud-ide.com. Έχοντας ήδη παρουσιάσει αυτά τα θέματα, θα εστιάσουμε στα εργαλεία που προσφέρει.

Version Control Systems

Το cloudbees παρέχει δύο εργαλεία διαχείρισης των εκδόσεων του κώδικα, το subversion και το git. Το subversion θεωρείται το de facto standard για παραδοσιακές πλατφόρμες ανάπτυξης, πχ in-house software development, ενώ το git είναι ένα σύγχρονο κατανεμημένο (distributed) σύστημα. Και στις δύο περιπτώσεις, ο κώδικας φιλοξενείται στα συστήματα του cloudbees απαλλάσσοντας τους χρήστες από την ανάγκη διαχείρισης.

Continuous Integration Server

Η πλατφόρμα παρέχει τον CI Server "Jenkins", ο οποίος αποτελεί τον διάδοχο του Hudson. Ο Jenkins είναι ίσως ο πιο διαδεδομένος Continuous Integration Server, με πάρα πολλά πρόσθετα. Συνεργάζεται με τα πιο δημοφιλή version control συστήματα και εκτελεί builds βασισμένα στο maven ή το ant. Η έκδοση που παρέχεται δεν έχει κάποιους περιορισμούς, υποστηρίζει τις ρυθμίσεις και τα πρόσθετα που υποστηρίζει μια εγκατάσταση in-house ενώ επιπλέον μπορεί να κάνει deployment της εφαρμογής απ ευθείας στην πλατφόρμα εκτέλεσης Run@Cloud του cloudbees.

Sonar

Πρόκειται για ένα εργαλείο ελέγχου της ποιότητας του κώδικα ως προς τους εξής άξονες: αρχιτεκτονική και σχεδιασμός, σχόλια, επαναλαμβανόμενα τμήματα, κώδικα ελέγχου (unit testing), πολυπλοκότητα, κανόνες συγγραφής και ροπή στα λάθη. Η αξία των στοιχείων αυτών στον πηγαίο κώδικα αναγνωρίζονται ευρέως πλέον και το sonar έχει αποκτήσει εξαιρετική δημοτικότητα, όντας απαραίτητο εργαλείο σε κάθε ομάδα.

Run@Cloud

Το περιβάλλον εκτέλεσης των εφαρμογών. Προς το παρόν υποστηρίζει μόνο εφαρμογές γραμμένες σε Java

Database - Cloudant

Πρόκειται για υπηρεσίες βάσεων δεδομένων. Η πρώτη προσφέρει σχεσιακές βάσεις δεδομένων, πιο συγκεκριμένα MySQL, ενώ η δεύτερη υπηρεσίες κατακευματισμένων, noSQL βάσεων, πιο συγκεκριμένα Apache CouchDB.

New Relic

Υπηρεσία παρακολούθησης (monitoring) και ανάλυσης των στοιχείων εκτέλεσης των εφαρμογών αλλά και των πόρων χρήσης του cloud.

Heroku

Το heroku παρέχει υπηρεσίες πλατφόρμας στο cloud για το Rack, το οποίο είναι ένα interface για την ανάπτυξη διαδικτυακών εφαρμογών σε Ruby. Επομένως, υποστηρίζει τη γλώσσα Ruby και το framework Ruby on Rails. Εκτός από το περιβάλλον εκτέλεσης προσφέρει πληθώρα noSQL βάσεων δεδομένων - μεταξύ αυτών Cloudant, Membase, MongoDB και Redis. Αναφορικά με τις υπηρεσίες σχεσιακών βάσεων δεδομένων η λύση που παρέχει είναι η Postgres. Τέλος, αξίζει να αναφερθεί ότι πρόσφατα παρουσίασε, σε συνεργασία με το Facebook, το Heroku Facebook App Package το οποίο προσφέρει εύκολη και γρήγορη ανάπτυξη εφαρμογών για την ομόθυμη υπηρεσία κοινωνικής δικτύωσης.

OpenShift

Το OpenShift είναι η πρόταση της Redhat στο χώρο του Platform as a Service cloud computing. Υποστηρίζεται επομένως από την μεγαλύτερη εταιρία λύσεων Linux, η οποία εκτός από την ομόνυμη διανομή αναπτύσσει και το JBoss, το δημοφιλέστερο open-source Java enterprise application server. Η πλατφόρμα φυσικά υποστηρίζει Java αλλά και Ruby, PHP, Python και Perl. Η συγκεκριμένη εταιρία επιπλέον προσφέρει το CirrAS, ένα εργαλείο για το αυτόματο deployment ενός jboss cluster στο cloud, το BoxGrinder, ένα εργαλείο για την αυτοποίηση δημιουργίας images για διάφορα Infrastructure as a Service clouds, το Steam Cannon ενώ υποστηρίζει ότι οι εφαρμογές για τον δικό της application server, το jboss μπορούν να τρέξουν ως έχουν στην πλατφόρμα OpenShift προστατεύοντας την εργασία που έχει ήδη γίνει.

Zend PHP Cloud Application Platform

Η πλατφόρμα αυτή απευθύνεται σε όσους υλοποιούν λύσεις σε PHP. Υπόσχεται PHP configuration consistency, session clustering, portability κ.α. Ανάμεσα στα εργαλεία που προσφέρει περιλαμβάνονται εργαλεία για monitoring, για τη διαχείριση του cluster, τη ρύθμιση των εφαρμογών, το deployment καθώς και ενοποίηση με το περιβάλλον ανάπτυξης zendstudio.

RightScale

Στο επίπεδο υπηρεσιών πλατφόρμας αξίζει να αναφέρουμε και την προσπάθεια της Rightscale. Πρόκειται για μια διαδικτυακή πλατφόρμα διαχείρισης ετερογενών clouds. Ουσιαστικά παρέχει τη δυνατότητα διαχείρισης δημοσίων, ιδιωτικών και υβριδικών clouds μέσα από ένα ενιαίο περιβάλλον. Συνργάζεται με εταιρίες παροχής λύσεων ιδιωτικών clouds όπως το Cloud.com και το eucalyptus διευκολύνοντας της διαχείρισή τους. Ταυτόχρονα επιτρέπει στους χρήστες να κάνουν migrate σε δημόσια clouds όπως το Amazon EC και το Rackspace.

Η RightScale προσφέρει ένα πρωτοποριακό τρόπο διαχείρισης ολόκληρων συστημάτων. Πρακτικά το deployment έχει μεταφερθεί στο επίπεδο συστημάτων. Ένα σύστημα μπορεί να αποτελείται από διαφορετικές εφαρμογές που τρέχουν σε διαφορετικούς servers ή και διαφορετικά συστήματα. Χρησιμοποιεί μια τεχνική στην οποία υπάρχουν έτοιμα Rightscale Server Templates και RightScale Server Scripts. Τα πρώτα ορίζουν το image, δηλαδή το λειτουργικό σύστημα και ίσως κάποιο βασικό λογισμικό ενώ τα δεύτερα ορίζουν το ρόλο του server και περιέχουν τις ρυθμίσεις του.

Συμπεράσματα

Ο χώρος των υπηρεσιών πλατφόρμας είναι ξεκάθαρα πιο περίπλοκος, με περισσότερες αλλά πιο εξειδικευμένες προτάσεις - αναμενόμενο αν αναλογιστούμε πως οι υπηρεσίες αυτές είναι ένα επίπεδο πιο πάνω από την υποδομή. Κάθε πάροχος στοχεύει σε πιο συγκεκριμένο κοινό, προσφέροντας περιβάλλοντα εκτέλεσης για ορισμένες γλώσσες μαζί με έναν αριθμό εργαλείων. Οι προτάσεις είναι ήδη αρκετές και αξιόλογες, από προτάσεις αρκετά πλήρεις όσον αφορά τις υπηρεσίες τους, προτάσεις που στοχεύουν σε προγραμματιστές αλλά και προτάσεις που προσπαθούν να μειώσουν το κόστος μετάβασης παρέχοντας περιβάλλοντα εκτέλεσης συμβατά με τα παραδοσιακά.

Μεταφορά στο cloud;

Είναι προφανές ότι ο χώρος του cloud αρχίζει να προσφέρει όλο και περισσότερες και ποιοτικότερες λύσεις σε πολλά ζητήματα. Πολλοί πάροχοι εμπλουτίζουν τις υπηρεσίες του ενώ νέες λύσεις εμφανίζονται τακτικά, όπως συμβαίνει με κάθε τεχνολογία. Όμως, κι εδώ έχουμε να αντιμετωπίσουμε ένα κοινό πρόβλημα, πού τελικά ταιριάζει η νέα τεχνολογία; ποια προβλήματα απ' όσα υπόσχεται μπορεί να λύσει και ποιες είναι οι παγίδες;

Σαν γενικός προσανατολισμός, θα μπορούσαμε να αναφέρουμε τα παρακάτω

Εργασίες επιλέξιμες προς μεταφορά στο cloud

1. Γενικές εργασίες που βρίσκονται ήδη σε τεχνολογίες virtualization
2. Συστήματα σε φάσης δοκιμών
3. Σουίτες εφαρμογών γραφείου, όπως email, on-line collaboration κ.α.
4. Τμήματα της ανάπτυξης λογισμικού
5. Διαδικασίες μαζικής επεργασίας με χαλαρούς περιορισμούς στην ασφάλεια
6. Μεμονωμένες εργασίες στις οποίες η χαμηλή ταχύτητα επικοινωνίας μεταξύ των επιμέρους συστημάτων δεν συνιστά πρόβλημα
7. Αποθηκευτικές λύσεις
8. Λύσεις αντιγράφων ασφαλείας

Εργασίες ανέτοιμες για το νέφος

1. Εργασίες σε ευαίσθητα δεδομένα
2. Σύνθετες εργασίες που περιλαμβάνουν τη συμμετοχή πολλών υποσυστημάτων και απαιτείται υψηλή ταχύτητα
3. Εργασίες με υψηλές απαιτήσεις σε καταγραφές
4. Εφαρμογές που χρησιμοποιούν λογισμικό τρίτων το οποίο έχει άδεια μη-συμβατή με το cloud computing [13]

Στον τομέα της ανάπτυξης λογισμικού, τα συστήματα που προσφέρονται παραδοσιακά έχουν ωριμάσει, προσφέρουν εκπληκτικές δυνατότητες - αδιανόητες μέχρι πριν λίγα χρόνια. Επιπλέον, έχουν αναπτυχθεί πολλαπλές υλοποιήσεις για κάθε τεχνολογία προσφέροντας πληθώρα επιλογών.

Στο κεφάλαιο αυτό θα επιχειρήσουμε να αποτιμήσουμε τις προσφερόμενες υπηρεσίες στο νέφος και να τις συγκρίνουμε με αυτές των παραδοσιακών συστημάτων. Φυσικά, οι τελευταίες υπερτερούν αφού υπάρχουν χρόνια, έχουν δοκιμαστεί και εμπλουτιστεί και είναι ωριμότερες, σε σχέση με τις πρώτες που εμφανίστηκαν μόλις πρόσφατα. Παρ' όλ' αυτά, θα επιχειρήσουμε

μία σύγκριση, κυρίως για να αναδείξουμε το τι μπορεί να μεταφερθεί στο νέφος και τι όχι - τουλάχιστον αυτή τη στιγμή.

Όπως ήδη έχουμε αναφέρει από το πρώτο κεφάλαιο, ο χώρος του cloud computing είναι μέχρι στιγμής εξαιρετικά ευμετάβλητος και οι αλλαγές διαδέχονται η μία την άλλη. Επομένως, οι παρακάτω συγκρίσεις και τα συμπεράσματα δεν πρέπει να θεωρούνται αειθαλή και αδιαμφισβήτητα. Όμως, σίγουρα, μια μελέτη όπως η παρακάτω είναι πάντα απαραίτητη για κάθε νέα τεχνολογία.

Συγγραφή κώδικα

Τα παραδοσιακά περιβάλλοντα ανάπτυξης προσφέρουν εξαιρετικές δυνατότητες. Συμπληρώνουν αυτόματα τον κώδικα (auto-complete), μας εμφανίζουν τις παραμέτρους που μπορεί να δεχτεί μια μέθοδος - ακόμα και για overloaded μεθόδους, προτείνουν αλλαγές στον κώδικα, κάνουν ορθογραφικό έλεγχο, συνδέονται με debuggers, με εξωτερικά εργαλεία καταγραφής σφαλμάτων και διαχείρισης έργων, "γράφουν" κώδικα για τυποποιημένες ρουτίνες κ.α.

Από την άλλη μεριά, τα διαδικτυακά περιβάλλοντα ανάπτυξης έχουν τις πολύ βασικές δυνατότητες, όπως μορφοποίηση του κώδικα, χρωματισμός και σε κάποιες περιπτώσεις συνεργάζονται με συστήματα ελέγχου εκδόσεων του κώδικα και build συστήματα. Βέβαια, υπερéχουν στον τομέα της εγκατάστασης και ρύθμισης, όμως ο χρόνος για ρύθμιση ενός συστήματος ανάπτυξης είναι εξαιρετικά μικρός ως προς το χρόνο χρήσης του.

Επομένως, τα διαδικτυακά περιβάλλοντα ανάπτυξης υπολείπονται δραματικά των αντίστοιχων παραδοσιακών, αφού τα πλεονεκτήματά τους είναι πολύ λιγότερο σημαντικά. Οι σημαντικές τους ελλείψεις καθώς και το γεγονός ότι κανένα δεν μπορεί να αποτελέσει μια ολοκληρωμένη λύση, καθιστούν την επαγγελματική ανάπτυξη εφαρμογών απαγορευτική.

Διαθέσιμες τεχνολογίες

Πλέον τα υπολογιστικά συστήματα είναι πολύπλοκα και πολλές τεχνολογίες έχουν εμφανιστεί ώστε να τα υποστηρίξουν. Κάποιες από τις τεχνολογίες αυτές έχουν μεταφερθεί ολικώς ή μερικώς στο νέφος, κάποιες άλλες όχι. Θα εξετάσουμε τι μπορεί να μεταφερθεί στο νέφος και σε ποιο βαθμό.

Databases

Τα συστήματα βάσεων δεδομένων είναι ίσως το πιο σημαντικό κομμάτι του επιχειρησιακού λογισμικού. Οι δυσκολίες του νέφους στην αποθήκευση των δεδομένων στο εκτελούμενο

instance στρέφουν τις λύσεις προς τους παρόχους υπηρεσιών πλατφόρμας. Εδώ λοιπόν, εκτός από την απώλεια της όποιας επένδυσης σε κλειστό λογισμικό πρέπει να τεθεί και το ζήτημα της διατήρησης και αξιοποίησης της υπάρχουσας εμπειρίας και τεχνογνωσίας από το προσωπικό.

Oracle

Η Oracle είναι ίσως το ευρύτερα χρησιμοποιούμενο εμπορικό σύστημα βάσης δεδομένων. Έχει πάρα πολλές δυνατότητες. Συνέπεια αυτού όμως είναι πως έχει και πολλά χαρακτηριστικά που δεν υπάρχουν στο standard (ansi-sql), όπως materialized views, analytical functions κλπ.

Κατά συνέπεια λοιπόν, εάν η εφαρμογή μας χρησιμοποιεί ή χρειάζεται αυτές τις δυνατότητες είναι προφανές ότι θα πρέπει να επιλέξουμε μια πλατφόρμα που θα προσφέρει βάσεις δεδομένων Oracle ως υπηρεσίες. Εναλλακτικά, θα πρέπει να υλοποιηθεί μια υβριδική λύση, με τη βάση δεδομένων on-premises και τη χρήση λύσεων όπως το Google Secure Data Connector για την ασφαλή ανταλλαγή δεδομένων ανάμεσα στην εφαρμογή και τη βάση δεδομένων. Επιπλέον, σε περίπτωση επιλογής δημοσίου νέφους ή εικονικού ιδιωτικού νέφους θα πρέπει να ληφθεί σοβαρά υπόψιν και η καθυστέρηση που θα εισάγει το δίκτυο στην επικοινωνία.

Postgresql - MySQL

Οι βάσεις αυτές είναι συστήματα που χρησιμοποιούνται κυρίως από εφαρμογές ανοιχτού λογισμικού, όπως είναι και οι ίδιες. Αυτό σημαίνει πως είναι πολύ πιο κοντά στην ansi sql και επομένως είναι αρκετά πιο πιθανό η μετάβαση στο νέφος να είναι πιο ομαλή, αρκεί να παρέχεται μια λύση κοντά στο stan-tarnt. Φυσικά, κι εδώ υπάρχουν τα νομικά και ασφαλιστικά ζητήματα αναφορικά με τα δεδομένα. Όμως, τεχνικά, αναμένεται να είναι πιο ομαλή η μετάβαση εδώ. Επιπλέον, λόγω της φύσης τους - ανοιχτό λογισμικό / λογισμικό ανοιχτού κώδικα - είναι πιο εύκολο να υιοθετηθεί από κάποιον πάροχο υπηρεσιών νέφους και αναμένεται ευρεία αποδοχή τους.

Message Broker Systems

Τα Message Broker Systems έχουν γίνει αρκετά δημοφιλή καθώς παρέχουν εξαιρετικές δυνατότητες ασύγχρονης επικοινωνίας. Έχουν γίνει αρκετά σύνθετα καθώς κάποια από αυτά ενσωματώνουν προηγμένες δυνατότητες. Επιπλέον, λόγω της ολοένα και μεγαλύτερης χρήσης κατανεμημένων συστημάτων η αξία και η δημοτικότητα τους μεγαλώνει. Επειδή, λοιπόν, είναι αρκετά πιθανό, αν όχι σίγουρο, πως μια επιχειρηματική εφαρμογή θα χρησιμοποιεί μια τέτοια τεχνολογία είναι απαραίτητο να αξιολογούμε το τι προσφέρει ο υποψήφιος πάροχος υπηρεσιών νέφους ώστε να γνωρίζουμε εάν η εφαρμογή μας μπορεί να μεταφερθεί στο νέφος και τι αλλαγές μπορεί να απαιτούνται.

ActiveMQ

Η ActiveMQ είναι ένα σύστημα που υλοποιεί το JMS Specification της Java. Εκτός όμως από τις προδιαγραφές αυτές παρέχει και επιπλέον δυνατότητες, όπως exclusive consumers, repeating

messages κλπ. Επιπρόσθετα, η αποθήκευση των μηνυμάτων μπορεί να γίνεται είτε στο σύστημα αρχείων είτε σε βάση δεδομένων. Η δεύτερη εκδοχή είναι αρκετά χρήσιμη, ώστε εάν το Message Broker που προσφέρει ο πάροχος δεν είναι συμβατό ή επαρκές για την εφαρμογή μας, μπορούμε να χρησιμοποιήσουμε ActiveMQ με τη σχεσιακή βάση του νέφους.

Caching

Καθώς οι εφαρμογές γίνονται όλο και πιο πολύπλοκες, ενώ αυξάνονται οι απαιτήσεις σε ταχύτητα, αυξάνεται και η ζήτηση για μηχανισμούς caching ώστε να επιταχύνεται η επεξεργασία. Πλέον, υπάρχουν αρκετές υλοποιήσεις ενώ έχουν αναπτυχθεί και συγκεκριμένα πρωτόκολλα επικοινωνίας με τις μηχανές caching. Οι υλοποιήσεις περιλαμβάνουν αυτόνομες εφαρμογές, κατανεμημένες, με χρήση μόνο μνήμης RAM ή συνδυασμό RAM και σκληρού δίσκου. Από την άλλη, αν και δεν υπάρχει κάποιο πρωτόκολλο που να υπερέχει ξεκάθαρα και να έχει γνωρίσει αποδοχή ανάλογη της SQL, αρκετοί κατασκευαστές ακολουθούν κάποιο από αυτά.

Memcached

Το memcached είναι ένα πρωτόκολλο για ανάγνωση και εγγραφή σε μηχανές caching. Αρχικά χρησιμοποιήθηκε από caching που χρησιμοποιούσαν μόνο μνήμη. Όμως, η διαφάνεια του μέσου αποθήκευσης (μνήμη ή δίσκος ή και τα δύο) του έδωσαν τη δυνατότητα να χρησιμοποιείται τόσο από caching συστήματα όσο και από noSQL βάσεις δεδομένων.

Application Servers

Όταν αναφερόμαστε σε application servers αυτομάτως μπορούμε να χρησιμοποιήσουμε και υπηρεσίες υποδομής εκτός από υπηρεσίες πλατφόρμας. Αυτό συμβαίνει γιατί οι application servers δεν απαιτούν υποχρεωτικά αποθήκευση δεδομένων ενώ μπορούν - τουλάχιστον οι πιο εξελιγμένοι - να συμμετάσχουν αυτόματα σε clusters. Αυτά σημαίνουν πως μπορούμε κάλλιστα να έχουμε ένα image με τον application server της επιλογής μας, το οποίο θα εκκινείται σε όσα instances απαιτείται. Στην περίπτωση μάλιστα αυτή μπορούμε να θεωρήσουμε πως η συμβατότητα είναι απόλυτα εξασφαλισμένη αφού στο image μπορούμε να έχουμε το λειτουργικό σύστημα της επιλογής μας καθώς και τον application server στην έκδοση που χρειαζόμαστε.

Στην περίπτωση των υπηρεσιών πλατφόρμας, αντίθετα, θα πρέπει να έχουμε ελέγξει ότι ο προσφερόμενος application server είναι συμβατός - αν όχι ο ίδιος - με αυτό που χρησιμοποιούμε ήδη ή που θέλουμε να χρησιμοποιήσουμε.

Java Application Servers

Οι application servers για εφαρμογές java διακρίνονται σε δύο κατηγορίες, τους Java Application Servers και του Java Enterprise Edition Application Server. Μάλιστα η oracle (sun) έχει

δημιουργήσει ένα εργαλείο το οποίο ελέγχει τη συμβατότητα του application server με τις προδιαγραφές που υποτίθεται ότι πληροί.

Έτσι, λοιπόν, δύο application server οι οποίοι πληρούν τις ίδιες προδιαγραφές μπορούν να εκτελέσουν την ίδια εφαρμογή χωρίς παραλλαγές, εφ' όσον η εφαρμογή δεν χρησιμοποιεί κάποια ειδικά χαρακτηριστικά του ενός εξ' αυτών αλλά περιορίζεται στα όρια των προδιαγραφών.

.NET

Καθώς η Microsoft δεν έχει δημιουργήσει προδιαγραφές για application servers της πλατφόρμας .net, δεν έχει ανοίξει ο δρόμος για δημιουργία εναλλακτικών servers στην πλατφόρμα αυτή. Επομένως, το μόνο που ενδιαφέρει σε αυτή την περίπτωση είναι η έκδοση της πλατφόρμας.

no-sql databases / Data warehouses

Σήμερα, υπάρχει μια στροφή από τις παραδοσιακές σχεσιακές βάσεις δεδομένων προς μη-σχεσιακές (ή βάσεις κλειδιού-τιμής - key-value databases) ή αλλιώς no-sql βάσεις δεδομένων. Κυριότερο χαρακτηριστικό αυτών είναι πως πρόκειται για κατανεμημένα συστήματα με πολύ συγκεκριμένους στόχους, για παράδειγμα, υψηλή ταχύτητα. Το πρόβλημα όμως είναι πως δεν έχουν καθοριστεί standards για τις γλώσσες των βάσεων αυτών.

Ανάμεσα στις κυριότερες βάσεις είναι η Cassandra, η HBase, η KyotoCabinet/KyotoTycoon, KumoFS κ.α. Οι βάσεις αυτές, αποκτούν επιπλέον δημοτικότητα επειδή μπορούν να αξιοποιηθούν και από/ως μηχανισμοί caching. Επιπρόσθετα, το γεγονός ότι είναι κατανεμημένα συστήματα εκ φύσεως τις κάνει άριστες επιλογές για το νέφος.

Στην περίπτωση λοιπόν που η εφαρμογή μας χρησιμοποιεί ή πρέπει να χρησιμοποιήσει μία ή περισσότερες από αυτές τις βάσεις δεδομένων θα πρέπει να προσέξουμε ότι αυτό που προσφέρεται από τον πάροχο υπηρεσιών νέφους - αν προσφέρεται - είναι συμβατό με αυτό το οποίο χρησιμοποιούμε ήδη. Συνήθως, οι πάροχοι, προσφέρουν μια τουλάχιστον τέτοια λύση, οπότε οφείλουμε να ελέγξουμε τουλάχιστον τη συμβατότητα.

Στον τομέα των αποθηκών δεδομένων τα πράγματα είναι σε πολύ πρώιμο στάδιο στο cloud computing. Αντίθετα, στα παραδοσιακά συστήματα έχουν αναπτυχθεί εξαιρετικά data warehouse software. Ένας από τους λόγους έχει να κάνει με την ιδιαίτερη αξία που έχουν τα συγκεντρωτικά δεδομένα για την επιχείρηση και ο ρόλος των data warehouses στη στρατηγική των επιχειρήσεων. Είναι επόμενο λοιπόν, να υπάρχει μια διστακτικότητα ως προς τη μεταφορά τους στο νέφος. Εν τούτοις, αναμένεται η εξέλιξη εδώ να είναι πολύ έντονη - έστω για χρήση σε ιδιωτικά clouds - καθώς οι αποθήκες δεδομένων έχουν αυξημένες απαιτήσεις τόσο σε αποθηκευτικούς χώρους όσο και σε υπολογιστική ισχύ, ενώ οι απαιτήσεις σε υπολογιστική ισχύ δεν είναι σταθερές, αλλά παρουσιάζουν απότομες αυξήσεις σε συγκεκριμένες χρονικές στιγμές.

Timers

Ένα άλλο πολύ σημαντικό χαρακτηριστικό των εφαρμογών είναι η δυνατότά τους να εκτελούν συγκεκριμένες εργασίες σε συσχετισμένες χρονικές στιγμές, επαναλαμβανόμενα ή μη. Η χρήση των timers λοιπόν είναι κάτι εξαιρετικά συνηθισμένο στις σύγχρονες εφαρμογές, ενώ έχουν αναπτυχθεί συστήματα με προχωρημένες δυνατότητες. Για το λόγο αυτό, οι πιο πολλοί πάροχοι υπηρεσιών νέφους προσφέρουν αντίστοιχες λειτουργίες ώστε και με περιορισμένες δυνατότητες.

Εάν θέλουμε να μεταφέρουμε την εφαρμογή μας στο νέφος, θα πρέπει να ελέγχουμε ότι οι δυνατότητες που χρειαζόμαστε είναι διαθέσιμες ή ότι μπορούμε να χρησιμοποιήσουμε το λογισμικό που επιθυμούμε.

Build Systems

Το build των εφαρμογών είναι μια διαδικασία που ενσωματώνει τη συγκέντρωση βιβλιοθηκών από το διαδίκτυο, το compile του κώδικα, εκτέλεση αυτοματοποιημένων ελέγχων, κ.α. Εφ' όσον οι τεχνολογίες που χρησιμοποιούμε είναι μεταφέρσιμες στο νέφος, τότε κατά πάσα πιθανότητα και το build θα μπορεί να λαμβάνει χώρα εκεί, αφού το ίδιο το build έχει ήδη μεταφερθεί εκεί.

Υπάρχουν ήδη αρκετοί πάροχοι που προσφέρουν build στο νέφος, πχ για την τη χρήση της πλατφόρμας Java, προσφέρονται λύσεις που χρησιμοποιούν το maven, ενώ δεν θα βρούμε εύκολα λύσεις για το Ant

Version Control Systems

Τα συστήματα ελέγχου εκδόσεων κώδικα, είναι κατεξοχήν δικτυακές εφαρμογές οι οποίες δεν απαιτούν συνεργασία με κάποιο άλλο σύστημα. Η διαθεσιμότητα τέτοιων συστημάτων είναι άριστη συμπεριλαμβάνοντας το subversion και το git. Επιπλέον, ακόμα κι αν ο πάροχος υποστηρίζει το ένα από τα δύο, υπάρχουν εργαλεία μετατροπής των δεδομένων από τη μια μορφή στην άλλη.

Επομένως αναφορικά με το ζήτημα των εκδόσεων έχουμε υψηλή προσφορά και συστήνεται η μεταφορά στο νέφος και είναι πολύ πιθανό να είναι εφικτή.

Testing

Η φύση των σεναρίων ελέγχου, δηλαδή μια μαζική πλήρως αυτοματοποιημένη εργασία τα καθιστά εξαιρετικούς υποψηφίους για μεταφορά στο νέφος. Αυτό που θα πρέπει να ελεγχθεί είναι η διαθεσιμότητα του επιθυμητού συστήματος πχ JUnit ή TestNG καθώς και το πιθανό κόστος αλλαγής. Φυσικά, για να μπορούν να εκτελεστούν τα σενάρια θα πρέπει να είναι γραμμένα σε γλώσσα και να απαιτούν τεχνολογίες που ήδη υποστηρίζονται.

Continuous Integration

Όπως έχουμε αναφέρει το CI ουσιαστικά αποτελεί μια διαφορετική προσέγγιση στο build των εφαρμογών. Οπότε, εάν ο πάροχος προσφέρει το build σύστημα που μας ενδιαφέρει, τότε είναι πολύ πιθανό να προσφέρει και continuous integration server συμβατό. Φυσικά, η εργασία που επιτελεί είναι μεταφέρσιμη στο νέφος και απόδειξη αυτού η διαθεσιμότητα. Θα βρούμε σχετικά εύκολα πλατφόρμας με υπηρεσίες CI για Hudson-Jenkins αντίθετα θα δυσκολευτούμε να βρούμε υποστήριξη για το Apache Continuum, ένα νέο σχετικά έργο με χαμηλή προς το παρόν δημοτικότητα.

Έλεγχος ποιότητας κώδικα

Εδώ, όπως και στην περίπτωση των version control systems, δεν απαιτείται η υποστήριξη κάποιων τεχνολογιών για την εφαρμογή μας. Ουσιαστικά γίνεται επεξεργασία των αρχείων κειμένου που αποτελούν τον κώδικα. Επομένως η εργασία του ελέγχου της ποιότητας του κώδικα είναι άμεσα μεταφέρσιμη στο νέφος και ήδη παρέχονται λύσεις για το Sonar.

QA & UAT

Οι ομάδες QA και UAT, χρησιμοποιούν εργαλεία παρακολούθησης σφαλμάτων (bug tracking systems, όπως bugzilla, jira κλπ) τα οποία προσφέρονται εδώ και χρόνια σαν υπηρεσίες λογισμικού.

Όμως, αναφορικά με την ανάπτυξη της εφαρμογής, έχουν την ιδιαιτερότητα ότι απαιτούν ξεχωριστά περιβάλλοντα. Το υπολογιστικό νέφος μπορεί να προσφέρει εξαιρετική βοήθεια στις ομάδες αυτές είτε σε επίπεδο υπηρεσιών πλατφόρμας είτε σε επίπεδο υπηρεσιών υποδομής.

Στην περίπτωση που η εφαρμογή μας μπορεί να μεταφερθεί σε υπηρεσίες πλατφόρμας, τότε η διευκόλυνση που παρέχεται είναι εξαιρετική, αφού οι πιο πολλοί πάροχοι δίνουν τη δυνατότητα εκτέλεσης διαφορετικών εκδόσεων της εφαρμογής ταυτόχρονα. Στην περίπτωση που η εφαρμογή δεν μπορεί να μεταφερθεί σε επίπεδο υπηρεσιών πλατφόρμας, μπορούν να αξιοποιηθούν υπηρεσίες από το επίπεδο υποδομής. Αν και αυτό δεν απέχει πάρα πολύ από τη

υφιστάμενη κατάσταση μπορεί να δώσει λύση στη διαχείριση των πόρων, ειδικά σε περιπτώσης stress test ή σε ξαφνικές απαιτήσεις για έλεγχο παλαιότερων εκδόσεων.

Συμπεράσματα

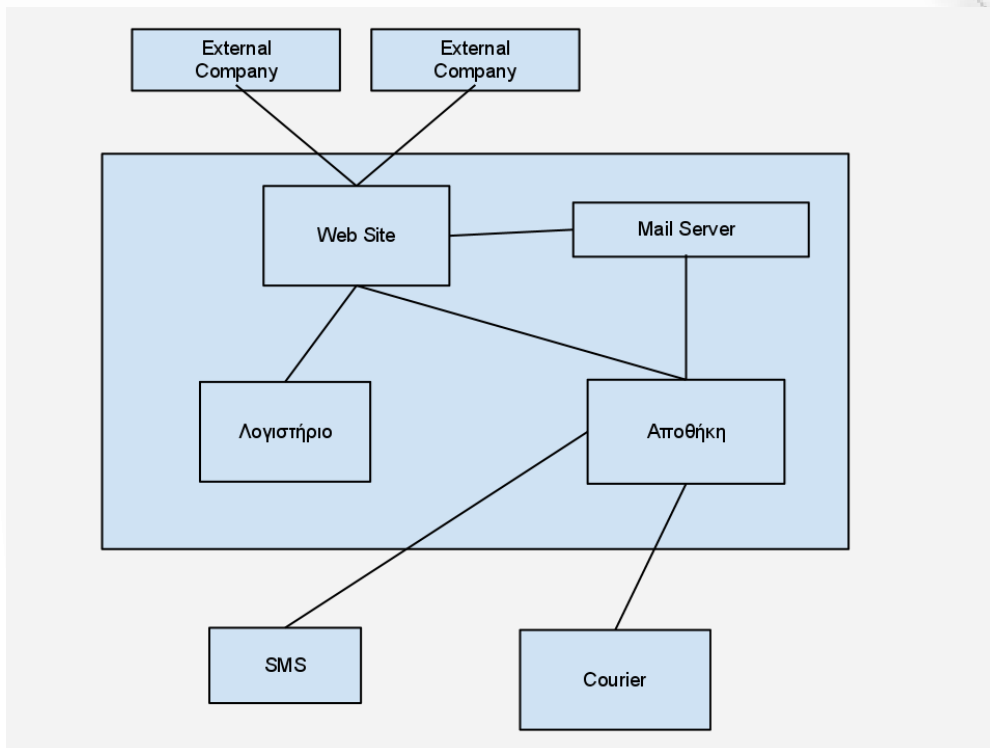
Γίνεται φανερό πως το υπολογιστικό νέφος δεν είναι μια λύση “plug ‘n’ play”. Θα πρέπει να ελέγχουμε τη συμβατότητα της εφαρμογής μας για να αποφασίσουμε τι μπορεί να μεταφερθεί στο νέφος. Μάλιστα, λόγω της ταχύτητας των αλλαγών στο χώρο του νέφους, ο έλεγχος θα πρέπει να γίνεται τη στιγμή που λαμβάνεται η απόφαση, αφού δεν είναι δυνατόν να υπάρξει ένα “πινακάκι” στο οποίο θα περιλαμβάνονται όλες οι τεχνολογίες και η συμβατότητα και διαθεσιμότητα στο νέφος.

Μελέτες - προτάσεις

Αξιοποίηση του cloud computing για την ανάπτυξη εφαρμογών

Στο σημείο αυτό θα παρουσιάσουμε το πώς πρακτικά θα μπορούσαμε να αξιοποιήσουμε το cloud για την ανάπτυξη μιας υποθετικής, αρκετά σύνθετης εφαρμογής ηλεκτρονικού εμπορίου. Η εφαρμογή αυτή θα λαμβάνει τις παραγγελίες των πελατών από το site, θα τις καταχωρεί σε ένα σύστημα ασύγχρονης επεξεργασία το οποίο θα ενημερώνει την αποθήκη και το λογιστήριο. Εάν όλα τα κομμάτια είναι διαθέσιμα θα επικοινωνεί με μια εξωτερική εταιρία courier για την παράδοση και θα ενημερώνει με sms τον πελάτη. Επιπλέον, κάθε αλλαγή στην κατάσταση της παραγγελίας θα γνωστοποιείται με email στον πελάτη. Το πληροφοριακό σύστημα της εταιρίας αποτελείται από τρία υποσυστήματα, το web site, την αποθήκη και το λογιστήριο ενώ επικοινωνεί μέσω web services με την εταιρία courier και μέσω http requests με την εταιρία υπηρεσίας αποστολής sms. Τέλος, μέσω web services που διαθέτει το web site, εξωτερικοί business-to-business πελάτες μπορούν να στέλνουν αυτόματα παραγγελίες.

Ένα γενικό διάγραμμα της εφαρμογής φαίνεται παρακάτω



Όπως μπορούμε ήδη να διακρίνουμε εμπλέκονται αρκετά υποσυστήματα καθένα από τα οποία θα πρέπει να αναπτύσσεται και να δοκιμάζεται τόσο μεμονωμένα όσο και σε συνεργασία με τα υπόλοιπα. Επομένως, εκτός από το περιβάλλον παραγωγής θα πρέπει να συντηρούνται περιβάλλοντα ανάπτυξης και ελέγχου, τόσο για το κάθε σύστημα ξεχωριστά όσο και για όλα μαζί.

Για την ανάπτυξη μπορεί να χρησιμοποιηθεί μια πλατφόρμα ανάπτυξης, πχ cloudbees. Έτσι, θα καλυφθούν οι ανάγκες σε version controll systems και continuous integration. Στην πλατφόρμα αυτή θα υπάρχουν και διάφορες βάσεις δεδομένων ανάλογα με το υποσύστημα, ανάλογα με την έκδοση και ανάλογα με το σκοπό, πχ testing. Με τον τρόπο αυτό θα είναι εφικτή και η εκτέλεση των αυτοματοποιημένων ελέγχων.

Περνώντας τώρα στις διαδικασίες που εμπλέκουν περισσότερο τον ανθρώπινο παράγοντα θα συγκρίνουμε τις παραδοσιακές προσεγγίσεις με αυτές που προτείνουμε για το cloud. Συνήθως, μια ομάδα ανάπτυξης ασχολείται με ένα υποσύστημα. Οι προγραμματιστές τρέχουν την εφαρμογή στο δικό τους υπολογιστή και για κάθε ομάδα ανάπτυξης διατίθεται και ένα μηχανήμα με τα υπόλοιπα υποσυστήματα. Αντίστοιχα συμβαίνει και για τις υπόλοιπες ομάδες, QA και UAT.

Οι υπηρεσίες cloud computing θα μπορούσαν να βοηθήσουν την εταιρία επιταχύνοντας δραστικά κάποιες διαδικασίες. Προτείνουμε λοιπόν τη χρήση υβριδικού cloud που θα

αποτελείται από εσωτερικό ιδιωτικό cloud βασισμένο στο eucalyptus και στο δημοσιο cloud της Amazon. Οδηγίες για την εγκατάσταση ιδιωτικού εσωτερικού cloud με το eucalyptus παρέχονται στο παράρτημα 1. Η εταιρία λοιπόν μπορεί να δημιουργήσει custom images όπως περιγράφεται στο παράρτημα 2 για το κάθε υποσύστημα. Κάθε image είτε θα περιέχει το απαραίτητο λογισμικό για την εκτέλεση των εφαρμογών και θα εγκαθίσταται χειροκίνητα η απαραίτητη έκδοση είτε μπορεί να διατηρεί ένα image για κάθε έκδοση. Στα επόμενα θα εξετάσουμε κάποια ρεαλιστικά σενάρια.

Σενάριο 1

Έστω ότι το web site τρέχει ταυτόχρονα σε παραπάνω από μια εκδόσεις επειδή κάποιοι εταιρικοί πελάτες δεν έχουν καταφέρει να αναβαθμίσουν τα συστήματά τους. Έτσι, η εταιρία είναι αναγκασμένη να διατηρεί περισσότερες από μία εκδόσεις του web site. Ας υποθέσουμε λοιπόν ότι διαπιστώνεται ένα πρόβλημα σε μια παλαιότερη έκδοση. Στις συνηθεις περιπτώσεις θα πρέπει να φορτωθεί η αντίστοιχη έκδοση της εφαρμογής, αντίστοιχες βάσεις δεδομένων και να γίνει έλεγχος. Εάν διαπιστωθεί πρόβλημα οι αντίστοιχες διαδικασίες πρέπει να επαναληφθούν για την ομάδα ανάπτυξης. Όλα αυτά είναι αρκετά χρονοβόρα και απαιτούν την εύρεση των πόρων, το στήσιμο βάσεων δεδομένων και των εφαρμογών και τη ρύθμισή τους.

Από την άλλη πλευρά, η λύση που προτείνουμε ελαχιστοποιεί τα προβλήματα. Η ομάδα ελέγχου μπορεί απλώς να εκκινήσει εικονικές μηχανές με τα αντιστοιχα images και να προβεί στους απαραίτητους ελέγχους. Στη συνέχεια, αν διαπιστωθεί σφάλμα στον κώδικα, η ομάδα ανάπτυξης μπορεί να εκκινήσει όσες εικονικές μηχανές απαιτούνται για τα κατάλληλα υποσυστήματα και να τα διατηρήσει μέχρι την επίλυση του προβλήματος. Στο τέλος, η λειτουργία των εικονικών μηχανών τερματίζεται ώστε να μην σπαταλώνται πόροι.

Σενάριο 2

Ας υποθέσουμε ότι πρέπει να γίνει stress test στην εφαρμογή ή σε κάποιο υποσύστημα. Αυτό συνήθως είναι μια επίπονη διαδικασία η οποία απαιτεί συνεργασία και οργάνωση ώστε να δεσμευτούν οι απαραίτητοι πόροι. Αντίθετα στο cloud μπορούν να χρησιμοποιηθούν εικονικές μηχανές οι οποίες θα έχουν προγράμματα που μιμούνται τους χρήστες. Φυσικά, με λίγα κλικ μπορούν να ξεκινήσουν ταυτόχρονα όσες τέτοιες εικονικές μηχανές είναι απαραίτητες. Επιπλέον, αυτό παρέχει τη δυνατότητα για την πιο συχνή διεξαγωγή τέτοιων ελέγχων, ακόμα και σε καθημερινή βάση.

Σενάριο 3

Κατά τη φάση της ανάπτυξης αποφασίζεται ότι πρέπει άμεσα να “βγει” μια νέα έκδοση ενός συγκεκριμένου υποσυστήματος, πχ γιατί η εταιρία courier αλλάζει το API που παρέχει ή γιατί

διαπιστώθηκε κάποιο σημαντικό σφάλμα. Αυτό θα σήμαινε ότι η ομάδα ανάπτυξης του υποσυστήματος της αποθήκης θα πρέπει να ακολουθήσει διαφορετικό κύκλο ανάπτυξης και να χρησιμοποιήσει τις παλαιότερες εκδόσεις των υπολοίπων υποσυστημάτων. Για να γίνει αυτό, το σύστημα με τις εφαρμογές web site και λογιστηρίου που χρησιμοποιείται ως integration περιβάλλον για την ομάδα ανάπτυξης της αποθήκης θα πρέπει να γυρίσει στην τρέχουσα έκδοση της παραγωγής από την έκδοση την οποία αναπτύσσεται αυτή τη στιγμή. Επιπλέον, είναι αρκετά δύσκολο να ασχοληθεί μερικώς η ομάδα με την αλλαγή που αφορά την εταιρία courier ή το σφάλμα.

Με τη χρήση του cloud η ευελιξία είναι πολύ μεγαλύτερη. Η ομάδα - ή μερικά μέλη της - εκκινούν εικονικές μηχανές με την απαιτούμενη έκδοση των άλλων υποσυστημάτων και εργάζονται κανονικά στις αλλαγές που απαιτούνται ενώ τα υπόλοιπα μέλη εργάζονται απεριόριστα στην τρέχουσα γραμμή ανάπτυξης. Η ευελιξία αυτή φυσικά μεταφέρεται τόσο στο QA όσο και του UAT..

Σενάριο 4

Το σενάριο αυτό ίσως το συναντήσουμε στο μέλλον, αφού αναφέρεται στην χρήση διαδικτυακών IDEs, αλλά είναι κάτι που και σήμερα ακόμα θα ήταν αρκετά χρήσιμο. Πολύ συχνά παρατηρείται ότι μια ομάδα έχει περισσότερο φόρτο από μια άλλη και το αντίθετο. Επομένως, θα ήταν σημαντικό να υπάρχει η εξής ευελιξία. Ανάλογα με το φόρτο εργασίας κάθε ομάδας ανάπτυξης να "μετακινούνται" προγραμματιστές, QA και UAT testers από την τρέχουσα ομάδα με το μικρότερο φόρτο προς αυτή με το μεγαλύτερο φόρτο. Κάτι τέτοιο, με τα σημερινά εργαλεία θα σήμαινε πως θα έπρεπε να συντηρούνται πολύ περισσότερες θέσεις εργασίας, ενώ το να έχει καθένας στον υπολογιστή του όλα τα εργαλεία στις απαραίτητες εκδόσεις ώστε να μπορεί να εργαστεί για οποιαδήποτε ομάδα ίσως να είναι και ανέφικτο λόγω διαφορών και πιθανών conflicts ανάμεσα στις εκδόσεις.

Αντίθετα, στα cloud IDEs αυτό είναι πολύ πιο εύκολο, αφού το κόστος συντήρησης των θέσεων εργασίας περιορίζεται αισθητά, ενώ ο καθένας έχει άμεσα το περιβάλλον που χρειάζεται κάνοντας απλώς διαφορετικό login στην αντίστοιχη ιστοσελίδα.

Χρήση User-Data

Τα user-data είναι δεδομένα που μπορούμε να περάσουμε σε ένα instance κατά την εκκίνησή του. Για την ακρίβεια είναι απλό κείμενο το οποίο μπορεί να περιέχει οτιδήποτε. Φυσικά η ερμηνεία των περιεχομένων του επαφίεται αποκλειστικά στο χρήστη. Με απλά λόγια, μπορεί κάποιος να δημιουργήσει ένα πρόγραμμα το οποίο θα περιμένει μεταβλητές περιβάλλοντος στα user-data, ενώ κάποιο άλλο πρόγραμμα μπορεί να περιμένει τα πακέτα που πρέπει να εγκατασταθούν, πχ ένα σε κάθε γραμμή ώστε να τα εγκαθιστά με την εκκίνηση του instance.

Επιπλέον, τα user-data μπορούν να αποτελούν ένα url που θα δείχνουν σε ένα αρχείο. Στην περίπτωση αυτή το instance μπορεί να κατεβάξει το αντίστοιχο αρχείο.

Η χρήση των user-data είναι εξαιρετικά υψηλής σημασίας, προσφέρει απίστευτη ευελιξία στη ρύθμιση του instance και μπορεί να αποτελέσει τη βάση για διάφορες μορφές λύσης στο θέμα της διαχείρισης των instances. Παρακάτω θα παρουσιάσουμε τρόπους αξιοποίησης αυτού του μηχανισμού.

User-Data ως αρχείο παραμέτρων

Αποτελεί κοινή πρακτική στην ανάπτυξη εφαρμογών η χρήση properties αρχείων που περιέχουν ρυθμίσεις για την εφαρμογή, όπως τα στοιχεία σύνδεσης στη βάση δεδομένων, τα στοιχεία συστημάτων ουρών κλπ. Σε αυτή την περίπτωση μπορούμε να έχουμε έτοιμα αρχεία ρυθμίσεων σε κάποιον δίσκο μας server και να ξεκινάμε κάθε instance δίνοντάς του το κατάλληλο αρχείο ρυθμίσεων - ως url. Με τον τρόπο αυτό μπορούμε να διαχειριζόμαστε εύκολα τις ρυθμίσεις που αφορούν περισσότερα από ένα instances.

User-data ως script

Η allestic.com έχει δημιουργήσει και ενσωματώσει στα image που παρέχει μια εφαρμογή η οποία ερμηνεύει τα user-data ως shell script και το εκτελεί. Με τον τρόπο αυτό μπορούμε να εγκαθιστούμε πακέτα εφαρμογών απαραίτητα για τις δικές μας εφαρμογές, να μεταφέρουμε από το δίκτυο έτοιμα αρχεία ρυθμίσεων και να τα αντιγράψουμε στο instance, ακόμα και να κατεβάζουμε ολόκληρες εφαρμογές, πχ jar αρχεία.

Για παράδειγμα λοιπόν, μπορούμε να έχουμε ένα image με έναν application server όπως το jboss. Το user-data κατεβάζει τα απαραίτητα jar αρχεία, τα αντιγράφει στον κατάλληλο κατάλογο στον application server και τον ξεκινά. Έτσι, με ένα μόνο custom image μπορούμε να τρέχουμε κάθε φορά την επιθυμητή εφαρμογή στην επιθυμητή έκδοση παρέχοντας το αντίστοιχο αρχείο.

Παραγόμενο αρχείο user-data

Στα παραπάνω σενάριο μπορούμε να προσθέσουμε τη δυνατότητα παραγωγής των αρχείων από ένα πρόγραμμα. Ας υποθέσουμε ότι έχουμε μια εφαρμογή η οποία μπορεί να παράγει αρχεία user-data. Τότε, περνάμε σαν user-data το url από το οποίο θα λάβει το αρχείο αυτό. Για να δώσουμε ένα πιο συγκεκριμένο παράδειγμα, θα παρουσιάσουμε μια εφαρμογή για την αυτοματοποίηση των deployments.

Η εφαρμογή αυτή θα έχει μια βάση δεδομένων στην οποία θα καταχωρούνται deployment profiles. Για παράδειγμα, ένα σύστημα αποτελείται από ένα σύστημα ουρών και δύο εφαρμογές, μία για την παραγωγή μηνυμάτων και μία για την κατανάλωση. Σε αυτό το σύστημα αντιστοιχεί

ένα profile, έστω για το QA και συγκεκριμένη έκδοση, το οποίο ορίζει αυτές τις οντότητες και τις μεταξύ τους σχέσεις. Με βάση αυτό το προφίλ μπορούμε να κάνουμε ένα deployment. Κατά την εκκίνηση των instances θα δημιουργηθούν δυναμικά τα user-data αρχεία βάση του ονόματος της ουράς.

Συμπεράσματα

Παρατηρώντας τον κόσμο του cloud computing διαπιστώνουμε έντονη δραστηριότητα, τόσο σε ερευνητικό όσο και σε επιχειρηματικό επίπεδο. Σχεδόν καθημερινά παρουσιάζονται νέες λύσεις. Στο επίπεδο υπηρεσιών υποδομής η κατάσταση είναι αρκετά ξεκάθαρη, δεδομένου ότι οι τεχνολογίες virtualization είναι διαδεδομένες και αρκετά ώριμες. Το ίδιο ισχύει και για τις υπηρεσίες λογισμικού (Software as a Service) μιας και διάφορες μορφές τους υπάρχουν στην αγορά αρκετά χρόνια. Αντίθετα, στο επίπεδο των υπηρεσιών πλατφόρμας ο χώρος παρουσιάζει εξαιρετική δυναμική μιας και υπάρχουν αρκετά κενά να καλυφθούν όσο και πολλές επιλογές στο πεδίο. Είναι αρκετές έως πολλές οι γλώσσες προγραμματισμού, ενώ τα εργαλεία που απαιτούνται σήμερα για την ανάπτυξη λογισμικού έχουν αυξηθεί.

Το cloud computing έρχεται με πολλές προοπτικές να προσφέρει λύσεις κυρίως στη διαχείριση των συστημάτων και να μειώσει την πολυπλοκότητα και το κόστος. Ταυτόχρονα όμως, όντας μια διαδικτυακή υπηρεσία επιφέρει ζητήματα ασφαλείας αλλά και διαθεσιμότητας. Εν τούτοις η δυναμική που έχει αναπτυχθεί πιστεύεται πως θα γεννήσει τις απαραίτητες απαντήσεις και μάλιστα σύντομα. Ενδεικτικό της ταχύτητας ανάπτυξης του πεδίου είναι πως κατά τη διάρκεια συγγραφής της εργασίας κάποιες από τις πληροφορίες ξεπεράστηκαν και έπρεπε να αλλαχθούν.

Η εργασία αυτή έγινε στο cloud. Για την ακρίβεια, εκτός από τις πλατφόρμες που χρησιμοποιήθηκαν για τις δοκιμές και τα παραδείγματα, η συγγραφή έγινε στο google docs. Στο ενδιάμεσο παρουσιάστηκαν δύο προβλήματα. Το πρώτο είχε να κάνει με τη σύνδεση στο internet η οποία παρουσίασε προβλήματα για κάποιες ώρες καθιστώντας ανέφικτη τη συνέχιση της εργασίας. Το δεύτερο είχε να κάνει με τη διαθεσιμότητα της υπηρεσίας, η οποία δεν λειτουργούσε για λίγα λεπτά. Άσχετα με την χρονική διάρκεια των προβλημάτων, αυτά κατέδειξαν επί του πρακτέου μία βασική αδυναμία των υπηρεσιών αυτών – την απόλυτη εξάρτησή τους από το δίκτυο.

Εν κατακλείδι, πιστεύουμε – με βάση την πορεία άλλων τεχνολογιών – ότι το cloud computing ωριμάζοντας θα μπορέσει να δώσει λύσεις, αφού όμως περάσει και από περιπτώσεις κακής ή μη κατάλληλης χρήσης οι οποίες θα βοηθήσουν στην καλύτερη και πιο ξεκάθαρη τοποθέτησή τους στις λύσεις πληροφορικής.

Παραρτήματα

Παράρτημα Ι. Εγκατάσταση ιδιωτικού cloud

Εισαγωγή

Θα δημιουργήσουμε ένα cloud με δύο μηχανήματα. Στο πρώτο, το Front-End (FE) , θα στήσουμε τα βασικά συστατικά του εκτός από το Node Controller (NC) το οποίο θα στηθεί στο δεύτερο μηχάνημα. Θα χρησιμοποιήσουμε το Ubuntu Server Edition 11 το οποίο έρχεται με το eucalyptus.

Εγκατάσταση Cloud Controller

Ξεκινάμε από το CD τη διαδικασία της εγκατάστασης και ρυθμίζουμε τις επιθυμητές επιλογές ή αφήνουμε τις προεπιλογές. Εννοείται ότι πρέπει να έχουμε συνδέσει το καλώδιου Ethernet με το switch. Σαν όνομα υπολογιστή μπορούμε να χρησιμοποιήσουμε το clic (από το Cloud controller). Στην συνέχεια μπορούμε να επιλέξουμε τον τρόπο εγκατάστασης του cloud. Στην οθόνη συμπλήρωσης της διεύθυνσης του cloud controller απλώς πατάμε enter. Ο οδηγός εγκατάστασης θα θεωρήσει ότι το μηχάνημα αυτό θα παίξει το ρόλο του cloud controller. Στην επόμενη οθόνη στην οποία επιλέγουμε τα συστατικά του cloud που θα χρησιμοποιήσουμε, αφήνουμε τις προεπιλογές, δηλαδή αποεπιλεγμένο το Node Controller και επιλεγμένα τα υπόλοιπα. Έπειτα συνεχίζουμε κανονικά τη διαδικασία εγκατάστασης.

Στο τέλος της εγκατάστασης, ρυθμίζουμε το FE. Δηλαδή του δίνουμε ένα FQDN (Fully Qualified Domain Name) για το e-mail. Τέλος, ορίζουμε το όνομα του πρώτου cluster, πχ cluster1 και το εύρος των διευθύνσεων (IP Range). Εδώ πρέπει να ορίσουμε μια σειρά από διευθύνσεις IP που θα χρησιμοποιούνται από τις εικονικές μηχανές. Αυτές οι διευθύνσεις μπορεί να είναι είτε εσωτερικές είτε εξωτερικές, αλλά σε κάθε περίπτωση να τηρούν τις εξής προϋποθέσεις:

- να είναι ελεύθερες
- να είναι στο ίδιο subnet με τον FE
- να είναι προσβάσιμες από αυτούς που πρόκειται να χρησιμοποιήσουν τις εικονικές μηχανές.

Για παράδειγμα, μπορούμε να βάλουμε το εύρος 192.168.0.200-192.168.0.220. Αυτό σημαίνει πρακτικά ότι μπορούμε να έχουμε μέχρι είκοσι εικονικές μηχανές ταυτόχρονα.

Εγκατάσταση Node Controller

Για την εγκατάσταση του node controller ακολουθούμε την παραπάνω διαδικασία με τις εξής δύο διαφορές:

1. συμπληρώνουμε τη διεύθυνση του cloud controller αν δεν εντοπιστεί αυτόματα από το πρόγραμμα εγκατάστασης
2. στην επιλογή πακέτων επιλέγουμε μόνο το Node controller

Φυσιολογικά το πρόγραμμα εγκατάστασης θα διαπιστώσει επίσης ότι έχουμε ήδη εγκαταστήσει ήδη ένα cluster controller και θα μας ενημερώσει με το μήνυμα "There is already a Eucalyptus cluster controller". Αν δεν εμφανιστεί αυτό το μήνυμα, το πρόγραμμα εγκατάστασης θα ζητήσει την IP διεύθυνση του controller.

Στη συνέχεια η εγκατάσταση θα προχωρήσει μέχρι να ολοκληρωθεί.

Ξεκινώντας

Κατ' αρχήν, για να αποφύγουμε προβλήματα και bugs κάνουμε ένα πλήρες upgrade σε κάθε μηχάνημα με τις εντολές

```
sudo su
apt-get update
apt-get upgrade
```

Στη συνέχεια, για να επιβεβαιώσουμε ότι όλα είναι εντάξει πηγαίνουμε στον cluster controller και δίνουμε την εντολή

```
cat /var/log/eucalyptus/registration.log
```

οπότε θα πρέπει να δούμε κάτι σαν το παρακάτω

```
2011-06-14 14:44:42+03.00 | 5366 -> Cluster cluster1 is already registered
2011-06-14 14:44:42+03.00 | 5368 -> Walrus 192.168.1.104 is already registered
2011-06-14 14:46:43+03.00 | 6944 -> Calling node cluster1 node 192.168.1.108
```

Η τελευταία γραμμή σημαίνει πως το node με IP 192.168.1.108 είναι ήδη μέλος του cluster, οπότε δεν έχουμε να κάνουμε register το node controller χειροκίνητα.

Εν συνεχεία, από ένα άλλο μηχάνημα του τοπικού δικτύου ανοίγουμε ένα browser και πηγαίνουμε στη διεύθυνση <https://192.168.1.104:8443>. Αποδεχόμαστε το security certificate που θα μας δείξει ο browser και συνδεόμαστε με όνομα χρήστη admin και κωδικό admin.

Αλλάζουμε τον default κωδικό και δίνουμε τη διεύθυνση email μας ώστε να λαμβάνουμε ειδοποιήσεις από το σύστημα.

Μέσα από το περιβάλλον αυτό μπορούμε να βρούμε τα certificates που απαιτούνται για να εργαζόμαστε καθώς και τα απαραίτητα κλειδιά και κωδικούς. Αφού τα κατεβάσουμε πρέπει να κατεβάσουμε και τα εργαλεία του euca2tools με την εντολή

```
sudo apt-get install euca2tools
```

και μετά να εκτελέσουμε την εντολή

```
euca-describe-availability-zones verbose
```

Η εντολή αυτή μας δείχνει τη διαθεσιμότητα των πόρων του cloud και θα πρέπει να δούμε κάτι σαν τον παρακάτω πίνακα

AVAILABILITYZONE	cluster1	192.168.1.104			
AVAILABILITYZONE	vm types	free/max	cpu	ram	disk
AVAILABILITYZONE	m1.small	0008 / 0008	1	192	2
AVAILABILITYZONE	c1.medium	0003 / 0003	1	256	5
AVAILABILITYZONE	m1.large	0002 / 0002	2	512	10
AVAILABILITYZONE	m1.xlarge	0002 / 0002	2	1024	20
AVAILABILITYZONE	c1.xlarge	0000 / 0000	4	2048	20

Αν στη στήλη free/max δούμε τα πάντα 0 σημαίνει είτε ο node controller δεν έχει γίνει register στο cluster είτε ότι υπάρχει κάποιο πρόβλημα με το hypervisor του node controller. Στο σύστημα της δοκιμής διαπιστώθηκε κάποιο bug στο bios της μητρικής με αποτέλεσμα να μην είναι δυνατή η ενεργοποίηση των virtualization extensions του επεξεργαστή. Το πρόβλημα λύθηκε με bios upgrade.

Επίσης, μέσα από την ιστοσελίδα, στην καρτέλα Store μπορούμε να βρούμε images για το cloud μας. Το σύστημα θα τα κατεβάσει και μετά θα τα εκτελέσει.

Παράρτημα II. Δημιουργία custom image

Θα περιγράψουμε συνοπτικά τη διαδικασία δημιουργία ενός ιδιωτικού debian-based AMI για το Amazon EC2. Ο ευκολότερος τρόπος για τη δημιουργία ενός custom image είναι να προσαρμόσουμε ένα υπάρχον image στις ανάγκες μας. Για παράδειγμα, μπορούμε να

χρησιμοποιήσουμε ένα image της alestic.com το οποίο περιλαμβάνει την έκδοση lenny της Debian. Αν και αυτή η έκδοση δεν είναι η τελευταία μπορούμε να την αναβαθμίσουμε εάν το χρειαζόμαστε. Θα περιγράψουμε τη διαδικασία τροποποίησης ενός υφιστάμενου image και τη χρήση του για να τρέξουμε instances.

Προαπαιτούμενα

Πρέπει να έχουμε δημιουργήσει ένα bucket στο οποίο θα ανεβάσουμε το image μας μετά την τροποποίηση. Το όνομα του bucket καλό είναι να περιλαμβάνει μόνο λατινικούς χαρακτήρες, διαφορετικά ίσως λάβουμε μηνύματα όπως το παρακάτω:

```
The specified bucket is not S3 v2 safe
```

Βήμα 1

Το πρώτο βήμα λοιπόν είναι να επιλέξουμε το image που θα αποτελέσει τη βάση για το δικό μας και να εκκινήσουμε μια εικονική μηχανή με αυτό.

Βήμα 2

Το επόμενο βήμα είναι να τροποποιήσουμε το image σύμφωνα με τις ανάγκες μας. πχ εγκαθιστούμε τον web server apache με την εντολή

```
apt-get install apache2
```

Βήμα 3

Στο τρίτο βήμα πρέπει να αντιγράψουμε από τον υπολογιστή μας το private key μας και το αντίστοιχο certificate στο υπάρχον image, στο φάκελο /mnt

```
scp -i <keypair_pem_file> <private_key_file> <certificate_file>  
<user@instance_public_dns>:/mnt
```

όπου:

- `keypair_pem_file`: Το pem αρχείο που αντιστοιχεί στο key pair με το οποίο ξεκινήσαμε το instance
- `private_key_file`: Το ιδιωτικό κλειδί του X.509 certificate
- `certificate_file`: Το cert αρχείο του X.509 certificate

- user: το όνομα χρήστη του image, όπως δηλώνεται από τον δημιουργό του image, στην περίπτωση μας root
- instance_public_dns: Το public dns του instance

Τα αρχεία μπορούμε να τα κατεβάσουμε από τις αντίστοιχες σελίδες της κονσόλας της Amazon. Επίσης, εκεί μπορούμε να βρούμε και τη διεύθυνση του instance.

Βήμα 4

Στο επόμενο βήμα δημιουργούμε το δικό μας image, με την εντολή:

```
ec2-bundle-vol -d /mnt -k /mnt/<private_key_file> -c /mnt/<certificate_file> -u <AWS_ACCOUNT_ID> -r <architecture>
```

όπου:

- AWS_ACCOUNT_ID: Το account id το οποίο μας έχει εκχωρήσει η amazon. Το βρίσκουμε στη σελίδα "Security credentials"
- architecture: η αρχιτεκτονική επεξεργαστή που απαιτεί το image για να εκτελεστεί, πχ i386, amd64

Βήμα 5

Στο προτελευταίο βήμα ανεβάζουμε το image που δημιουργήσαμε στο bucket μας με την παρακάτω εντολή:

```
ec2-upload-bundle -b <bucket_name> -m /mnt/image.manifest.xml -a <aws_access_key> -s <aws_secret_key>
```

όπου:

- bucket_name το όνομα του bucket που δημιουργήσαμε στην αρχή
- aws_access_key το access key ID μας και
- aws_secret_key το secret access key μας.

Όπως και τα προηγούμενα μπορούμε να τα βρούμε στη σελίδα του amazon console

Βήμα 6

Το τελευταίο βήμα είναι να καταχωρήσουμε το image μας. Για αυτό, μας είναι απαραίτητα η jana6, το unzip και το arj-tools, τα οποία μπορούμε να εγκαταστήσουμε με τις παρακάτω εντολές


```
apt-get install sun-java6-jdk unzip
```

```
wget http://s3.amazonaws.com/ec2-downloads/ec2-api-tools.zip
```

```
unzip -d /opt ec2-api-tools.zip
```

```
export EC2_HOME=/opt/ec2-api-tools/bin
```

```
export JAVA_HOME=/usr/lib/jvm/java-6-sun
```

Τέλος, καταχωρούμε το image με την εντολή

```
ec2-register <bucket_name>/image.manifest.xml -k <private_key_file> -c <certificate_file>
```

Εάν κάνουμε κάποιο ορθογραφικό λάθος, πιθανόν να δούμε ένα μήνυμα όπως :

Client.InvalidManifest: HTTP 403 (Forbidden) response from URL

http://s3.amazonaws.com:80/<bucket_name>/image.manifest.xml: check your S3 ACLs are correct

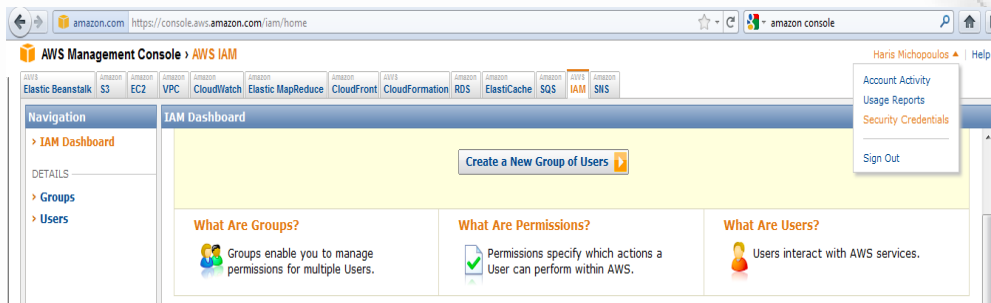
Παράρτημα III: Πρώτα βήματα στο Amazon EC2

Αφού εγγραφούμε στο cloud της Amazon θα μας εκχωρηθούν:

- ένα AWS Account ID
- ένα Canonical User ID
- ένα Access Key
- ένα X.509 Certificate
- ένα Key Pair για το CloudFront.

Μετά την εγγραφή μας μπορούμε να χρησιμοποιήσουμε τις υπηρεσίες του EC2 αφού γίνουμε πρώτα συνδρομητές σε κάθε μία από αυτές, δεν μας εκχωρείται δηλαδή αυτόματα το δικαίωμα χρήσης όλων των υπηρεσιών. Βέβαια, δεν υπάρχει κάποιο κόστος ενεργοποίησης, αντίθετα η χρέωση βασίζεται στη χρήση.

Αφού λοιπόν εγγραφούμε και ενεργοποιήσουμε τις επιθυμητές υπηρεσίες, μπορούμε να δούμε τα παραπάνω στοιχεία στη σελίδα "security credentials" που είναι διαθέσιμη από την κονσόλα της Amazon.



Στη σελίδα που θα ανοίξει μπορούμε να δούμε τα στοιχεία μας.

Access Key

Σύμφωνα με τη σελίδα τα Access Keys χρησιμοποιούνται για να κάνουμε REST ή Query requests στο API οποιουδήποτε AWS service. Αυτό πρακτικά σημαίνει πως εάν θέλουμε να διαχειριστούμε προγραμματιστικά τους πόρους μας στο Amazon EC2 θα πρέπει να χρησιμοποιούμε στα προγράμματά μας αυτά τα στοιχεία, όπως τα βρίσκουμε στη σελίδα των security credentials.

Access Credentials

There are three types of access credentials used to authenticate your requests to AWS services: (a) access keys, (b) X.509 certificates, and (c) key pairs. Each access credential type is explained below.

Access Keys | X.509 Certificates | Key Pairs

Use access keys to make secure REST or Query protocol requests to any AWS service API. We create one for you when your account is created — see your access key below.

Your Access Keys

Created	Access Key ID	Secret Access Key	Status
May 15, 2011	AKIAJ5YXDWTMLRD2GWQA	Show	Active (Make Inactive)
June 19, 2011	AKIAJF3FXBDS6JND72DQ		

For your protection, you should never share your secret access keys. AWS best practice recommends frequent key rotation.

[Learn more about Access Keys](#)

Έτσι, λοιπόν, στο παράδειγμα του κεφαλαίου "Εργαλεία ανάπτυξης για το Amazon's EC2", θα πρέπει να έχουμε ένα αρχείο με το όνομα "AwsCredentials.properties" και τα εξής στοιχεία

```
ACCESS_KEY_ID=AKIAJ5YXDWTMLRD2GWQA
SECRET_KEY=use the corresponding Secret key
```

Επιπλέον, επειδή τα προγράμματα που χρησιμοποιούμε, όπως τα `ec2_tools` ή τα `euca_tools` ουσιαστικά κάνουν κλήσεις στα αντίστοιχα APIs, πρέπει να τους δίνουμε αυτά τα στοιχεία (είτε στη γραμμή εντολών ως παραμετρο, είτε ως μεταβλητές περιβάλλοντος, είτε σε αρχεία) για να μπορούν να λειτουργήσουν σωστά.

Key Pairs

Τα `key pairs` είναι ζεύγη κλειδιών σαν αυτά που χρησιμοποιούμε για να κάνουμε πιστοποίηση χρηστών χωρίς χρήση κωδικού στο `ssh` και η χρήση για την οποία προορίζονται είναι ακριβώς αυτή, δηλαδή η σύνδεσή μας στα `instances` όπως και άλλες λειτουργίες `ssh` όπως το `scp` (`secure copy`). Αυτό που πρέπει να προσέξουμε είναι ότι για λόγους ασφαλείας η Amazon δεν διατηρεί το ιδιωτικό κλειδί στους `server` της ούτε για ζεύγη που έχουμε δημιουργήσει από την κονσόλα της, επομένως είμαστε εμείς υπεύθυνοι γι αυτό. Η πιο απλή λύση είναι να ανεβάσουμε το δικό μας δημόσιο κλειδί ώστε να κάνουμε πολύ εύκολα `login` εάν είμαστε σε Linux.

Regions και Availability Zones

Πριν προχωρήσουμε στα παραδείγματα, πρέπει να αναφέρουμε ένα άλλο σημαντικό στοιχείο: τις περιοχές (`region`) και τις ζώνες διαθεσιμότητας. Πρόκειται για γεωγραφικές περιοχές στις οποίες έχει καταφέρει πόρους η Amazon. Για λόγους ταχύτητας και απόδοσης συχνά διάφορες υπηρεσίες και πόροι θα πρέπει να βρίσκονται στην ίδια ζώνη ώστε να χρησιμοποιήσει το ένα το άλλο. Για παράδειγμα, δεν έχει νόημα να έχουμε ένα `volume` σε περιοχή των Ηνωμένων Πολιτειών και να το προσαρτήσουμε σε ένα `instance` που τρέχει στην περιοχή της Ευρώπης, αφού η καθυστέρηση στην εγγραφή των δεδομένων θα ήταν απογοητευτική. Επομένως, προκειμένου να μην αντιμετωπίσουμε εκπλήξεις, θα πρέπει να επιλέγουμε την περιοχή στην οποία θα εργαστούμε και να δημιουργούμε στην ίδια περιοχή αυτά που σκοπεύουμε να χρησιμοποιήσουμε μαζί.

Instance User Data

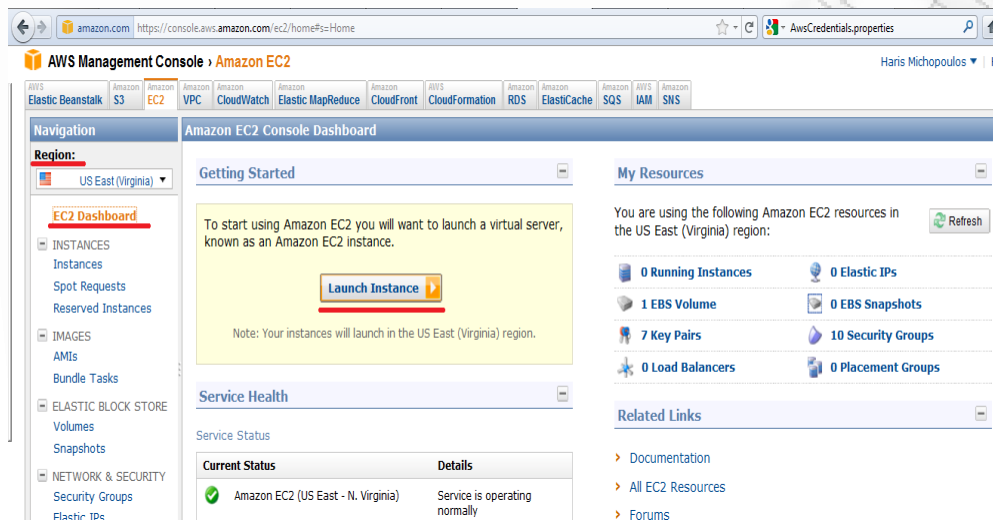
Κατά την εκκίνηση ενός `instance` έχουμε τη δυνατότητα να περάσουμε παραμέτρους σε αυτό. Τα δεδομένα αυτά μπορούμε να τα χρησιμοποιήσουμε ως παραμέτρους για τη ρύθμιση του `instance` και των εφαρμογών που τρέχουν σε αυτό.

Εκκίνηση Instance

Για να ξεκινήσουμε ένα `instance` στο Amazon EC2 πρέπει γενικά: να επιλέξουμε το κατάλληλο `image` καθώς και τις προδιαγραφές του εικονικού μηχανήματος στο οποίο θα τρέχει, να

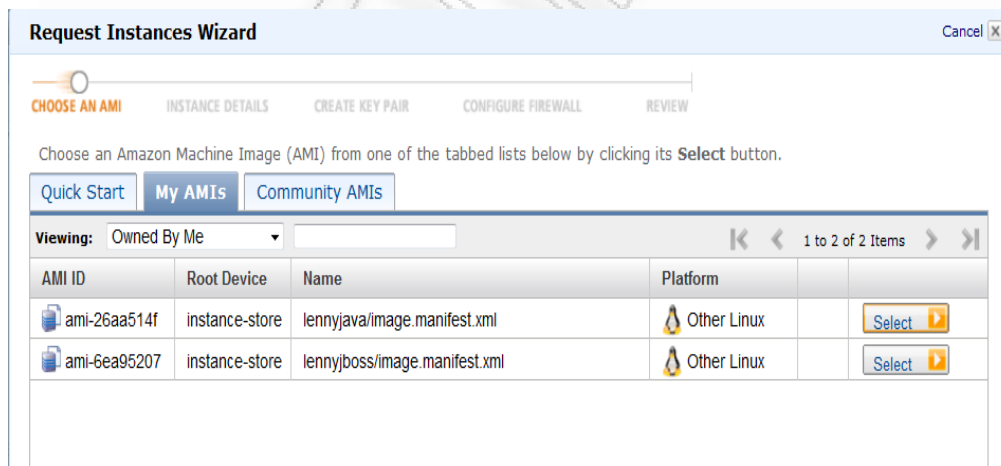
αντιστοιχίσουμε το επιθυμητό key pair και, να ορίσουμε χαρακτηριστικά ασφαλείας, όπως το ποιες πόρτες θα είναι ανοιχτές.

Πηγαίνουμε στην κονσόλα, στο tab EC2 <https://console.aws.amazon.com/ec2/home>, αν θέλουμε αλλάζουμε την περιοχή, και επιλέγουμε Launch Instance από το EC2 Dashboard.



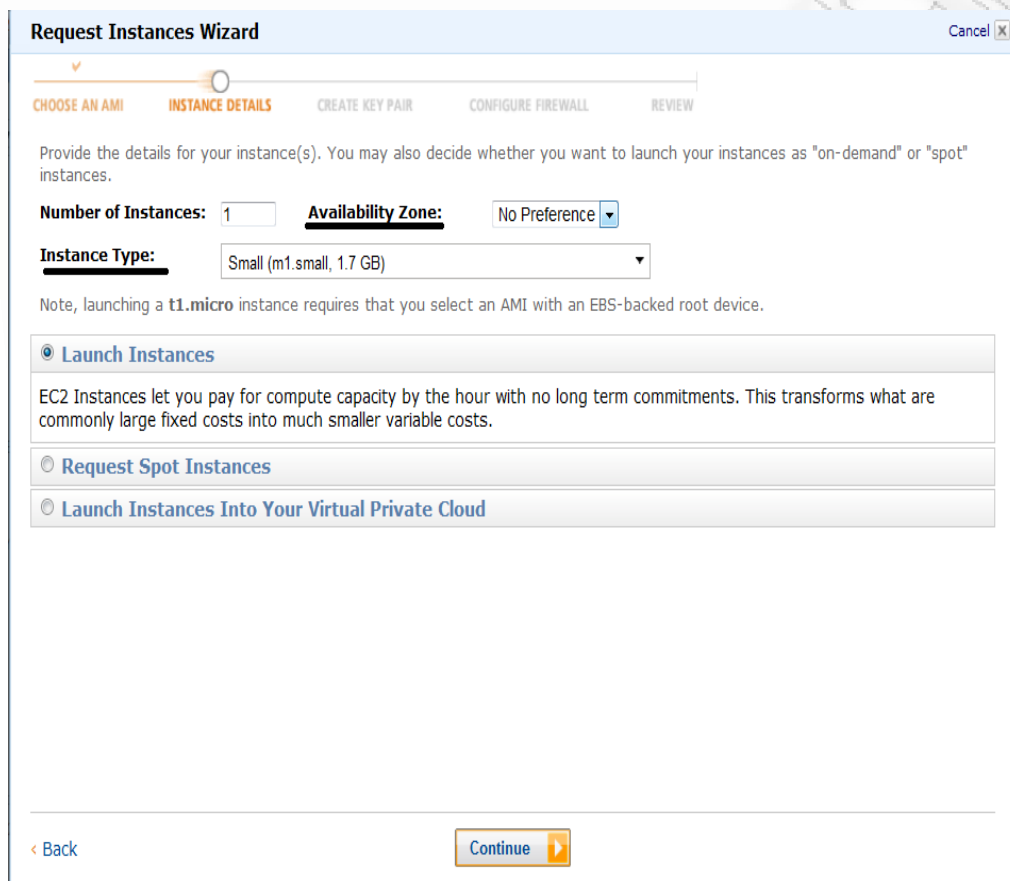
Επίσης, στο δεξιό μέρος της οθόνης βλέπουμε και ποιες υπηρεσίες έχουμε ενεργοποιήσει και τι χρησιμοποιούμε στη συγκεκριμένη περιοχή.

Πατώντας το Launch Instance εμφανίζεται ο οδηγός, στο πρώτο βήμα του οποίου πρέπει να επιλέξουμε το επιθυμητό image.



Εδώ βλέπουμε τα δικά μας images, αυτά δηλαδή που έχουμε δημιουργήσει εμείς. Στην καρτέλα Community AMIs μπορούμε να βρούμε πάρα πολλά κοινόχρηστα images. Επιλέγουμε το επιθυμητό image και πατάμε “Select”

Στην επόμενη οθόνη επιλέγουμε Availability Zone και τις προδιαγραφές της εικονικής μηχανής



The screenshot shows the 'Request Instances Wizard' window, specifically the 'INSTANCE DETAILS' step. The progress bar at the top indicates the current step. Below the progress bar, there is a text prompt: 'Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.' The form contains the following fields: 'Number of Instances' set to 1, 'Availability Zone' set to 'No Preference', and 'Instance Type' set to 'Small (m1.small, 1.7 GB)'. A note below the fields states: 'Note, launching a t1.micro instance requires that you select an AMI with an EBS-backed root device.' Below the note are three radio button options: 'Launch Instances' (selected), 'Request Spot Instances', and 'Launch Instances Into Your Virtual Private Cloud'. At the bottom of the window, there are 'Back' and 'Continue' buttons.

Πατώντας Continue, θα μας εμφανιστεί η επόμενη οθόνη στην οποία μπορούμε να επιλέξουμε κάποια πιο προχωρημένα χαρακτηριστικά, καθώς και να ορίσουμε τα “User Data”. Μία από τις χρήσεις των user data είναι να αποτελούν scripts ή εντολές που θα εκτελεστούν κατά την εκκίνηση του instance ακριβώς μετά την εκκίνηση των υπηρεσιών αυτού πχ web server, mail server κλπ και μπορούν να αξιοποιηθούν για την αρχικοποίηση του instance. Τα user data μπορούν να είναι απλό κείμενο με τις εντολές είτε ένα script αρχείο. Στο παράδειγμα, με την εκκίνηση του instance θα τρέξει η εντολή echo που θα γράψει το PAPER στο αρχείο /test.file.

Θα πρέπει βέβαια να σημειωθεί πως τα user-data απλώς παρέχονται στο instance. Η ερμηνεία τους και η αξιοποίησή τους επαφίονται αποκλειστικά σε αυτό. Επομένως, δεν θα τρέξουν όλα τα instances την εντολή αυτή παρά μόνο όσα είναι ρυθμισμένα.

Request Instances Wizard Cancel

CHOOSE AN AMI **INSTANCE DETAILS** CREATE KEY PAIR CONFIGURE FIREWALL REVIEW

Number of Instances: 1
Availability Zone: No Preference

Advanced Instance Options

Here you can choose a specific **kernel** or **RAM disk** to use with your instances. You can also choose to enable CloudWatch Detailed Monitoring or enter data that will be available from your instances once they launch.

Kernel ID: Use Default **RAM Disk ID:** Use Default

Monitoring: Enable CloudWatch detailed monitoring for this instance
(additional charges will apply)

User Data: echo PAPEI > /test.file

as text
 as file base64 encoded

Termination Protection: Prevention against accidental termination.

[Back](#) [Continue](#)

Πατάμε Continue. Στην επόμενη οθόνη μας δίνεται η δυνατότητα να ορίσουμε κάποια ζεύγη κλειδιών-τιμών που θα αντιστοιχούν σε αυτό το instance. Αυτά μπορούμε να τα χρησιμοποιήσουμε για την ευκολότερη διαχείριση των instance μας.

Request Instances Wizard Cancel X

CHOOSE AN AMI | **INSTANCE DETAILS** | CREATE KEY PAIR | CONFIGURE FIREWALL | REVIEW

Add tags to your instance to simplify the administration of your EC2 infrastructure. A form of metadata, tags consist of a case-sensitive key/value pair, are stored in the cloud and are private to your account. You can create user-friendly names that help you organize, search, and browse your resources. For example, you could define a tag with key = Name and value = Webserver. You can add up to 10 unique keys to each instance along with an optional value for each key. For more information, go to [Using Tags](#) in the *EC2 User Guide*.

Key (127 characters maximum)	Value (255 characters maximum)	Remove
Name	Pa.Pei. DS Example	X
		X

[Add another Tag.](#) (Maximum of 10)

Συνεχίζουμε πατώντας Continue. Στην επόμενη οθόνη μπορούμε να δημιουργήσουμε ένα νέο ζεύγος κλειδιών, να χρησιμοποιήσουμε ένα υφιστάμενο ή να προχωρήσουμε χωρίς ζεύγος κλειδιών. Η τελευταία επιλογή έχει νόημα εάν γνωρίζουμε τα στοιχεία σύνδεσης (username / password) αλλιώς δεν θα μπορούμε να συνδεθούμε στο instance.

Request Instances Wizard Cancel X

CHOOSE AN AMI | INSTANCE DETAILS | **CREATE KEY PAIR** | CONFIGURE FIREWALL | REVIEW

Public/private key pairs allow you to securely connect to your instance after it launches. To create a key pair, enter a name and click **Create & Download your Key Pair**. You will then be prompted to save the private key to your computer. Note, you only need to generate a key pair once - not each time you want to deploy an Amazon EC2 instance.

Choose from your existing Key Pairs

Your existing Key Pairs*: ▼

Create a new Key Pair

Proceed without a Key Pair

Πατάμε Continue για να συνεχίσουμε στη ρύθμιση του Firewall. Εκεί θα πρέπει να ανοίξουμε τις απαραίτητες θύρες. **Προσοχή!** Η πορτα του ssh (22) δεν ανοίγει αυτόματα. Θα πρέπει να τη δηλώσουμε εδώ, αλλιώς δεν θα μπορούμε να συνδεθούμε!

Request Instances Wizard Cancel X

CHOOSE AN AMI INSTANCE DETAILS CREATE KEY PAIR **CONFIGURE FIREWALL** REVIEW

Security groups determine whether a network port is open or blocked on your instances. You may use an existing security group, or we can help you create a new security group to allow access to your instances using the suggested ports below. Add additional ports now or update your security group anytime using the Security Groups page.

Choose one or more of your existing Security Groups

Create a new Security Group

Group Name

Group Description

Inbound Rules

Create a new Custom TCP rule

rule:

Port range:

(e.g., 80 or 49152-65535)

Source:

(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

TCP Port (Service)	Source	Action
80 (HTTP)	0.0.0.0/0	Delete
90	0.0.0.0/0	Delete
22 (SSH)	0.0.0.0/0	Delete

[Back](#)

Στην τελευταία καρτέλα μπορούμε να κάνουμε προεπισκόπηση των επιλογών μας και να εκκινήσουμε το instance πατώντας Launch.

Εδώ πρέπει να κάνουμε την εξής σημείωση. Εάν διαχειριστούμε το firewall από το "Security Group" της καρτέλας "Amazon EC2", σε αντίθεση με την καρτέλα στον οδηγό εκκίνησης instance, θα πρέπει μετά τις αλλαγές να πατήσουμε "Apply". Το κουμπί αυτό δεν φαίνεται άμεσα και θα πρέπει να μεγαλώσουμε το παράθυρο για να το δούμε. Αλλιώς δεν θα ισχύουν οι αλλαγές. Επίσης, οι αλλαγές δεν επηρεάζουν instances που τρέχουν ήδη.

Spot Requests
Reserved Instances

- IMAGES
 - AMIs
 - Bundle Tasks
- ELASTIC BLOCK STORE
 - Volumes
 - Snapshots
- NETWORK & SECURITY
 - Security Groups**
 - Elastic IPs
 - Placement Groups
 - Load Balancers
 - Key Pairs

1 Security Group selected

Security Group: papei

Details **Inbound***

Create a new Custom TCP rule

rule:

Port range:

(e.g., 80 or 49152-65535)

Source:

(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

TCP Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete
80 (HTTP)	0.0.0.0/0	Delete
90	0.0.0.0/0	Delete
34	0.0.0.0/0	Delete

Your changes have not been applied yet.

Έχοντας λοιπόν ξεκινήσει το instance μπορούμε να δούμε πληροφορίες γι αυτό και να το διαχειριστούμε στην καρτέλα Amazon EC2 -> Instances.

The screenshot shows the AWS Management Console interface for Amazon EC2. The 'My Instances' section is active, displaying a table with one instance: 'Pa.Pei. DS Example' (Instance ID: i-0cdf6e6c, AMI ID: ami-26aa514f, Type: m1.small, Status: running). Below the table, the details for the selected instance are shown in a key-value format:

Property	Value
AMI	Unnamed (ami-26aa514f)
Security Groups	papei
Status	running
VPC ID	-
Source/Dest. Check	-
Placement Group	-
RAM Disk ID	ari-42b95a2b
Key Pair Name	debian-al
Monitoring	basic
Elastic IP	-
Zone	us-east-1a
Type	m1.small
Owner	684049668834
Subnet ID	-
Virtualization	paravirtual
Reservation	r-aa4ce2c4
Platform	-
Kernel ID	aki-6eaa4907
AMI Launch Index	0
Root Device	-

Αν κάνουμε δεξί κλικ στο instance, μπορούμε να επιλέξουμε connect. Στο παράθυρο που θα ανοίξει μας παρουσιάζονται οι οδηγίες για τη σύνδεση στο instance μέσω ssh.

The dialog box titled 'Connect Help - Secure Shell (SSH)' provides instructions for connecting to the instance. The instructions are as follows:

1. Open the SSH client of your choice (e.g., PuTTY, terminal).
2. Locate your private key file, `debian-al.pem`
3. Use `chmod` to make sure your key file isn't publicly viewable, `ssh` won't work otherwise:
`chmod 400 debian-al.pem`
4. Connect to your instance using its Public DNS [`ec2-50-16-40-182.compute-1.amazonaws.com`].

Example
Enter the following command line:

```
ssh -i debian-al.pem root@ec2-50-16-40-182.compute-1
```

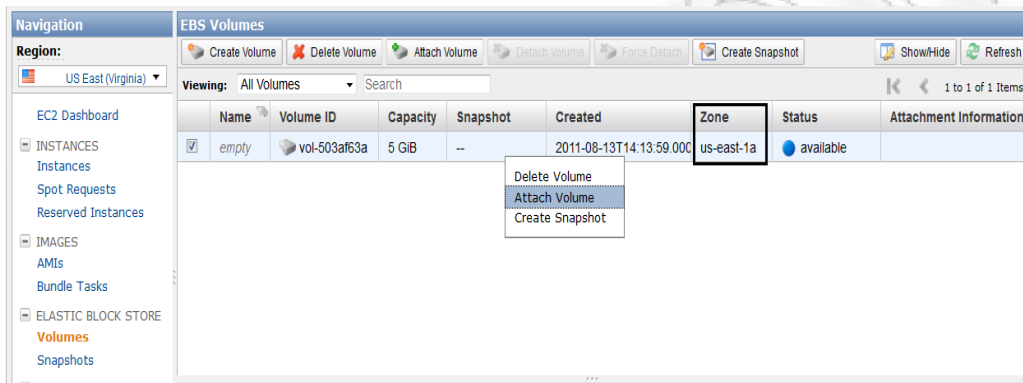
Having problems?

Close

Επομένως, θα πρέπει να έχουμε το ιδιωτικό κλειδί, να του έχουμε δώσει τα κατάλληλα δικαιώματα (βήμα 3) και να εκτελέσουμε την εντολή από τον κατάλογο στον οποίο έχουμε αποθηκευμένο το ιδιωτικό κλειδί.

Attach Volume

Όπως έχουμε αναφέρει ένα volume είναι ένας εικονικός δίσκος τον οποίο μπορούμε να προσαρτήσουμε σε ένα instance. Για να γίνει αυτό πάμε στο Volumes της καρτέλας Amazon EC2. Επιλέγουμε το volume και κάνουμε δεξί κλικ -> Attach Volume

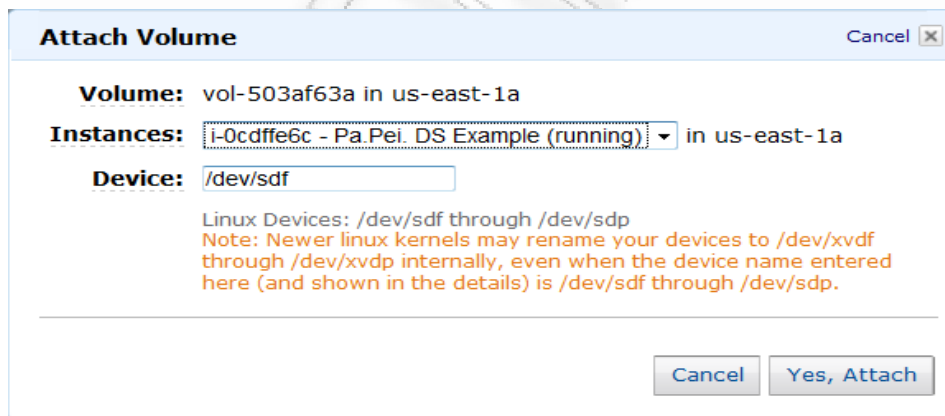


The screenshot shows the Amazon EC2 console interface for EBS Volumes. The 'Zone' column of the volume table is highlighted, and a context menu is open with 'Attach Volume' selected. The volume details shown are: Name: empty, Volume ID: vol-503af63a, Capacity: 5 GiB, Snapshot: --, Created: 2011-08-13T14:13:59.000, Zone: us-east-1a, Status: available.

Name	Volume ID	Capacity	Snapshot	Created	Zone	Status	Attachment Information
empty	vol-503af63a	5 GiB	--	2011-08-13T14:13:59.000	us-east-1a	available	

Εδώ πρέπει να προσέξουμε ότι το volume είναι στο ίδιο zone με το instance. Διαφορετικά δεν θα μπορούσαμε να το προσαρτήσουμε.

Πατάμε Attach Volume και μας εμφανίζεται το επόμενο παράθυρο



The 'Attach Volume' dialog box shows the following information:

- Volume:** vol-503af63a in us-east-1a
- Instances:** i-0cddf66c - Pa.Pei. DS Example (running) in us-east-1a
- Device:** /dev/sdf

Linux Devices: /dev/sdf through /dev/sdp
Note: Newer linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Buttons: Cancel, Yes, Attach

Επιλέγουμε το instance (εδώ βλέπουμε και την τιμή που είχαμε δώσει στο κλειδί Name κατά την εκκίνηση του instance). Επίσης, μπορούμε να ορίσουμε το device στο οποίο θα φαίνεται ο

εικονικός δίσκος. Μετά την προσάρτηση, το χρησιμοποιούμε όπως ακριβώς έναν δίσκο στον υπολογιστή μας, δηλαδή μπορούμε να κάνουμε format, mount κλπ.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

Βιβλιογραφία

1. http://en.wikipedia.org/wiki/Cloud_computing
2. Virtualization, Cloud Computing - Technologies and Strategies of the Ubiquitous Data Center, Brian J.S. Chee and Curtis Franklin, Jr., CRC Press, ISBN: 978-1-4398-0612-8
3. Introduction to cloud computer architecture, White paper, 1st Edition, June 2009, Sun microsystems
4. <http://www.eucalyptus.com/>
5. Cloud computer for dummies, Judith Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper, Wiley Publishing, Inc., ISBN: 978-0-470-48470-8
6. Cloud Computing Principles, Systems And Applications, Nick Antonopoulos, Lee Gillam, Springer, ISBN 978-1-84996-240-7
7. Cloud Computing and Software Services Theory and Techniques, Syed A. Ahson, Mohammad Ilyas, CRC Press, ISBN: 978-1-4398-0316-5
8. Handbook of Cloud Computing, Borko Furht, Armando Escalante, Springer, ISBN 978-1-4419-6523-3
9. http://en.wikipedia.org/wiki/Economies_of_scale
10. webinar "Why Cloud Management Makes Sense", RightScale.com
11. Cloud Computing : How the CIO becomes a hero, Joerg Meinke, 2009, IBM

Περιεχόμενα

[Αντί προλόγου - Σκοπός](#)

[Εισαγωγή στο Cloud Computing](#)

[Εισαγωγή](#)

[Ορολογία](#)

[Τι είναι το cloud computing](#)

[Ιστορικά στοιχεία](#)

[Χαρακτηριστικά μιας υποδομής cloud](#)

[Ελαστικότητα και κλιμάκωση](#)

[Αυτο-τροφοδοσία \(Self-service provisioning\)](#)

[Application Programming Interfaces \(API\)](#)

[No vendor lock-in](#)

[Μέτρηση και χρέωση υπηρεσιών](#)

[Παρακολούθηση απόδοσης](#)

[Ασφάλεια](#)

[Αρχιτεκτονική του Cloud](#)

[Ταξινόμηση μοντέλων παροχής υπηρεσιών Cloud](#)

[Infrastructure as a Service](#)

[Platform as a Service](#)

[Software as a Service](#)

[Hardware as a Service](#)

[Ταξινόμηση συστημάτων Cloud βάση τοπολογίας](#)

[Δημόσια συστήματα cloud](#)

[Ιδιωτικά \(εσωτερικά\) συστήματα cloud](#)

[Εικονικά ιδιωτικά συστήματα cloud](#)

[Υβριδικά συστήματα cloud](#)

[Το οικονομικό μοντέλο του cloud computing](#)

[Συμπεράσματα](#)

[Το οικοσύστημα ανάπτυξης λογισμικού](#)

[Εισαγωγή](#)

[Εργαλεία ανάπτυξης](#)

[Build systems](#)

[Dependency management](#)

[Testing: QA & UAT](#)

[Συμπεράσματα](#)

[Πλατφόρμες συγγραφής κώδικα στο cloud](#)

[Εισαγωγή](#)

[Παρουσίαση](#)

[Codility](#)

[Kodingen](#)

[Cloudbees & could-ide.com](#)
[Ανάπτυξη με το eclipse](#)
[Salesforce.com](#)
[Συμπεράσματα](#)
[Infrastructure as a Service](#)
[Βασικές υπηρεσίες υποδομής](#)
[CPU and RAM](#)
[Storage](#)
[Databases](#)
[Queues](#)
[Amazon's Elastic Compute Cloud \(EC2\)](#)
[Υπηρεσίες του Amazon's EC2](#)
[AWS Elastic Beanstalk](#)
[Amazon S3](#)
[Amazon VPC](#)
[Amazon CloudWatch](#)
[Elastic Load Balancing](#)
[Elastic Block Store \(ESB\)](#)
[Amazon CloudFront](#)
[AWS CloudFormation](#)
[Amazon Elastic MapReduce](#)
[Amazon Relational Database Service \(RDS\)](#)
[Amazon ElasticCache](#)
[Amazon Simple Notification Service](#)
[Amazon Simple Queue Service \(SQS\)](#)
[AWS Identity and Access Management \(IAM\)](#)
[Εργαλεία ανάπτυξης για το Amazon's EC2](#)
[Eucalyptus](#)
[Αρχιτεκτονική](#)
[Συστατικά του eucalyptus](#)
[OpenNebula](#)
[Συμπεράσματα](#)
[Platform as a Service](#)
[Βασικές υπηρεσίες υποδομής](#)
[Google App Engine](#)
[Υπηρεσίες του Google App Engine](#)
[Runtime environments](#)
[Go Runtime Environment](#)
[Python Runtime Environment](#)
[Java Runtime Environment](#)
[Datastore](#)
[Google Accounts](#)
[Caching](#)
[Scheduled Tasks](#)

[Task Queues](#)
[Integrated services](#)
[Ανάπτυξη για το Google App Engine](#)
[AppScale](#)
[CloudBees](#)
[Version Control Systems](#)
[Continuous Integration Server](#)
[Sonar](#)
[Run@Cloud](#)
[Database - Cloudant](#)
[New Relic](#)
[Heroku](#)
[OpenShift](#)
[Zend PHP Cloud Application Platform](#)
[RightScale](#)
[Συμπεράσματα](#)
[Μεταφορά στο cloud:](#)
[Συγγραφή κώδικα](#)
[Διαθέσιμες τεχνολογίες](#)
[Databases](#)
[Postgresql - MySQL](#)
[Message Broker Systems](#)
[Caching](#)
[Application Servers](#)
[no-sql databases / Data warehouses](#)
[Timers](#)
[Build Systems](#)
[Version Control Systems](#)
[Testing](#)
[Continuous Integration](#)
[Έλεγχος ποιότητας κώδικα](#)
[QA & UAT](#)
[Συμπεράσματα](#)
[Μελέτες - προτάσεις](#)
[Αξιοποίηση του cloud computing για την ανάπτυξη εφαρμογών](#)
[Σενάριο 1](#)
[Σενάριο 2](#)
[Σενάριο 3](#)
[Σενάριο 4](#)
[Χρήση User-Data](#)
[User-Data ως αρχείο παραμέτρων](#)
[User-data ως script](#)
[Παραγόμενο αρχείο user-data](#)
[Συμπεράσματα](#)

Παράρτημα

Παράρτημα I. Εγκατάσταση ιδιωτικού cloud

Εισαγωγή

Εγκατάσταση Cloud Controller

Εγκατάσταση Node Controller

Ξεκινώντας

Παράρτημα II. Δημιουργία custom image

Προαπαιτούμενα

Βήμα 1

Βήμα 2

Βήμα 3

Βήμα 4

Βήμα 5

Βήμα 6

Παράρτημα III: Πρώτα βήματα στο Amazon EC2

Access Key

Key Pairs

Regions και Availability Zones

Instance User Data

Εκκίνηση Instance

Attach Volume

Βιβλιογραφία

Περιεχόμενα