

Cloud Computing

από την πλευρά του ελέγχου

Επαλήθευση, Επικύρωση και Συντήρηση Λογισμικού Εαρινό Εξάμηνο 2021
Οικονομικό Πανεπιστήμιο Αθηνών

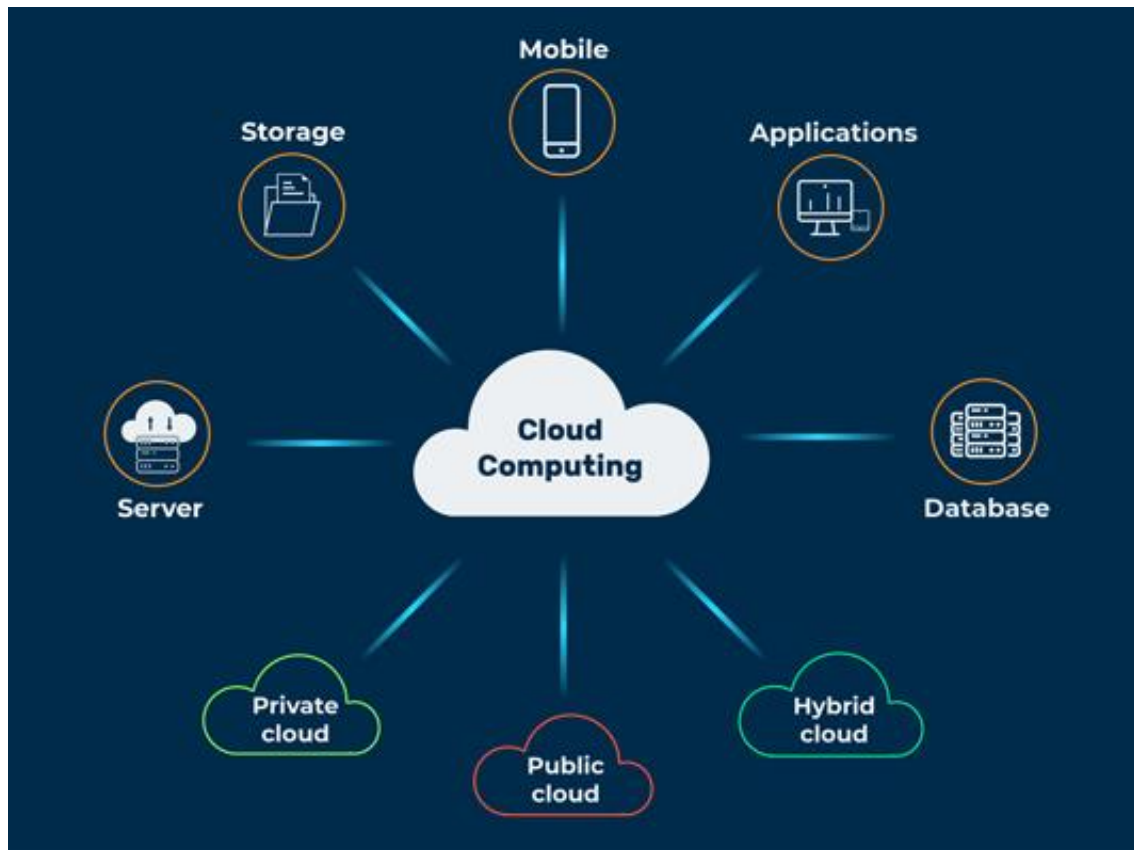
Κουζούμπαση Θεμελίνα, p3170076
Σουβατζιδάκη Κωνσταντίνα, p3170149

Περιεχόμενα

1. Εισαγωγή: Ορισμός Νέφους και Μοντέλα Ανάπτυξης.....	1
1.1 Τι είναι το νέφος.....	1
1.2 Μοντέλα ανάπτυξης.....	2
2. Υπολογιστική Νέφους: Ορισμός, Αρχιτεκτονική και Υπηρεσίες.....	5
2.1 Cloud vs Cloud Storage vs Cloud Computing.....	5
2.2 Αρχιτεκτονική.....	5
2.3 Μοντέλα υπηρεσιών νέφους.....	7
3. Έλεγχοι Λογισμικού στο Νέφος: Ορισμός και Τύποι.....	10
3.1 Τι είναι ο έλεγχος λογισμικού.....	10
3.2 Τι είναι Οι Έλεγχοι στο Νέφος.....	11
3.3 Τύποι ελέγχου στο νέφος βάσει οντότητας – στόχου.....	11
4. Τεχνικές Ελέγχων στο Νέφος.....	12
4.1 Λειτουργικός έλεγχος (Functional Testing).....	12
4.2 Μη λειτουργικός έλεγχος (Non-Functional Testing).....	13
4.3 Έλεγχος Δυνατοτήτων (Ability Testing).....	15
5. Τα οφέλη των Ελέγχων σε Περιβάλλον Νέφους.....	16
5.1 Η ανάγκη για γρήγορους ελέγχους λογισμικού στην σημερινή εποχή:.....	16
5.2 Τα πλεονεκτήματα των υπηρεσιών νέφους για την αυτοματοποίηση των ελέγχων λογισμικού:.....	16
6. Τα Οφέλη του Νέφους για τους Εξ' Αποστάσεως Ελέγχους (Remote Testing).....	19
6.1 Η ανάγκη για εξ' αποστάσεως πραγματοποίηση ελέγχων λογισμικού.....	19
6.2 Προβλήματα της εξ' αποστάσεως πραγματοποίησης ελέγχων.....	20
7. Οι Δυσκολίες των Ελέγχων σε Περιβάλλον Νέφους.....	22
7.1 Οι προκλήσεις των ελέγχων λογισμικού στο νέφος.....	22
7.2 Πότε το νέφος δεν είναι η κατάλληλη επιλογή.....	25
8. Η Σωστή Κατεύθυνση Προετοιμασίας Μετάβασης στο Νέφος για Ελέγχους... ..	25
8.1 Η ανάγκη για σωστή οργάνωση των έργων λογισμικού.....	25
8.2 Βήματα για την σωστή προετοιμασία αυτοματοποίησης των ελέγχων με υπηρεσίες νέφους.....	27
9. Εργαλεία Ελέγχου στο Νέφος.....	29
9.1 Δημοφιλή εργαλεία ελέγχου στο νέφος.....	29
10. Σύνοψη.....	30
11. Βιβλιογραφία.....	31

1. Εισαγωγή: Ορισμός Νέφους και Μοντέλα Ανάπτυξης

Το **νέφος (cloud)** είναι ένας πολύ δημοφιλής όρος που πρωτοεμφανίστηκε το 1950 και χρησιμοποιείται παγκοσμίως στο χώρο της πληροφορικής.



Το Νέφος
(Πηγή Εικόνας: Google)

1.1 Τι είναι το νέφος

Πρόκειται για ένα δίκτυο από **εξυπηρετητές (servers)** που λειτουργεί σαν ένας οργανισμός, με τον κάθε έναν από αυτούς να έχει μία συγκεκριμένη μοναδική λειτουργία.

Οι **εξυπηρετητές** αυτοί, όπως επίσης και το **λογισμικό** και οι **βάσεις δεδομένων** που «τρέχουν» σε αυτούς, είναι προσβάσιμοι μόνο μέσω του Διαδικτύου.

Οι λειτουργίες των εξυπηρετητών δεν είναι συγκεκριμένης κατηγορίας, αλλά διευρύνονται σε μια πληθώρα κατηγοριών. Συγκεκριμένα, η αποθήκευση και διαχείριση δεδομένων, η εκτέλεση εφαρμογών και η παροχή διαφόρων υπηρεσιών ή περιεχομένου είναι κάποιες από τις κατηγορίες λειτουργιών, τις οποίες μπορεί να παρέχει ένας εξυπηρετητής στο νέφος.

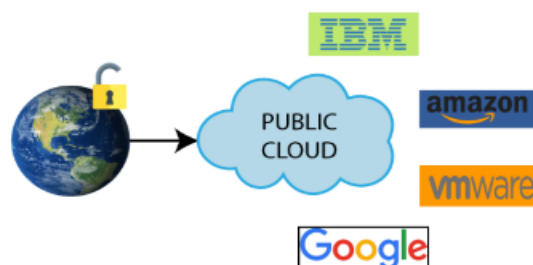
1.2 Μοντέλα ανάπτυξης

Ως **μοντέλο ανάπτυξης** ορίζεται ένα μοντέλο το οποίο περιγράφει το περιβάλλον στο οποίο εγκαθίστανται οι εφαρμογές και υπηρεσίες του νέφους, έτσι ώστε να είναι διαθέσιμες στους χρήστες.

Τα πιο γνωστά μοντέλα ανάπτυξης είναι τα ακόλουθα:

☁ **Δημόσιο (Public)**

Το Δημόσιο Νέφος (public cloud) είναι διαθέσιμο προς όλους, προκειμένου να αποθηκεύουν και να έχουν πρόσβαση σε πληροφορίες μέσω του Διαδικτύου. Η χρέωση που υποστηρίζει βασίζεται στο μοντέλο χρέωση-ανά-χρήση (**pay-per-use**), και γι' αυτό το λόγο είναι ιδανικό για επιχειρήσεις μικρής ή μεσαίας κλίμακας με μικρό προϋπολογισμό.



Δημόσιο Νέφος

(Πηγή εικόνας: <https://www.javatpoint.com/types-of-cloud>)

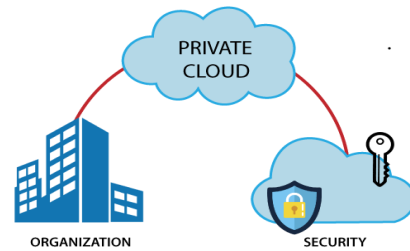
Παραδείγματα: Amazon EC2, IBM SmartCloud Enterprise, Microsoft

Τα υπέρ και κατά του Public Cloud:

- | | |
|--|---|
| ✓ εύκολη κλιμακωσιμότητα | - μη κατάλληλο για ευαίσθητα δεδομένα από άποψη ασφάλειας |
| ✓ πρόσβαση χωρίς γεωγραφικούς περιορισμούς | - η απόδοση εξαρτάται από την απόδοση της σύνδεσης με τον πάροχο του νέφους |
| ✓ χαμηλό κόστος | - ο χρήστης δεν έχει τον έλεγχο των δεδομένων του |
| ✓ υψηλή αξιοπιστία | |
| ✓ εύκολη διαχείριση | |
| ✓ απεριόριστοι χρήστες | |

☁️ **Ιδιωτικό (Private)**

Το Ιδιωτικό Νέφος (private cloud) είναι διαθέσιμο μόνο στην επιχείρηση στην οποία ανήκει μέσω Διαδικτύου ή εικονικού προσωπικού δικτύου (VPN), και στεγάζεται είτε στην ίδια την επιχείρηση είτε σε κάποια άλλη με την οποία συνεργάζεται. Η αυτονομία που παρέχει όσο αφορά τη χωρητικότητα και το δίκτυο αλλά και το υψηλό επίπεδο ασφάλειας το καθιστά ιδανικό για επιχειρήσεις μεγαλύτερης κλίμακας, ενώ δεν συστήνεται για επιχειρήσεις μικρότερης κλίμακας λόγω του κόστους στέγασης του cloud σε εξωτερικό συνεργάτη.



Ιδιωτικό Νέφος

(Πηγή εικόνας:

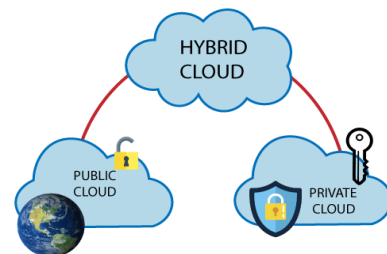
<https://www.javatpoint.com/types-of-cloud>)

Τα υπέρ και κατά του Private Cloud:

- | | |
|--|-------------------------|
| ✓ υψηλό επίπεδο ασφάλειας | - περιορισμένη πρόσβαση |
| ✓ μεγαλύτερος έλεγχος στο server | βάσει τοποθεσίας |
| ✓ αναπροσαρμογή βάσει απαιτήσεων | - απαιτείται ειδικός |
| ✓ υψηλή απόδοση ταχύτητας, χωρητικότητας | πληροφορικής |
| ✓ γρήγορη κατανομή υπολογιστικών πόρων κατ' απαίτηση | - περιορισμένος αριθμός |
| ✓ πλήρης αυτονομία | χρηστών |

☁️ **Υβριδικό (Hybrid)**

Το Υβριδικό Νέφος (Hybrid cloud) αποτελεί συνδυασμό του Δημοσίου και του Ιδιωτικού Νέφους, δηλαδή παρέχει την δυνατότητα διαχωρισμού υπηρεσιών και δεδομένων βάσει της ασφάλειας που απαιτούν. Είναι ιδανικό για επιχειρήσεις που επωφελούνται από τα πλεονεκτήματα του Δημοσίου Νέφους, αλλά χρειάζονται περισσότερη ασφάλεια από αυτή που παρέχει.



Υβριδικό Νέφος

(Πηγή εικόνας:

<https://www.javatpoint.com/types-of-cloud>)

Παραδείγματα: Google Application Suite (Gmail, Google Apps, Google Drive), Office 365 (MS Office στον Ιστό, One Drive), Amazon Web Services

Τα υπέρ και κατά του Hybrid Cloud:

- | | |
|---|--|
| ✓ υψηλότερο επίπεδο ασφάλειας από το public cloud | - χαμηλότερο επίπεδο ασφάλειας από το private cloud |
| ✓ ταχύτητα στην παροχή νέων προϊόντων και υπηρεσιών | - περίπλοκη διαχείριση |
| ✓ προσαρμοστικότητα | - η αξιοπιστία των υπηρεσιών εξαρτώνται απ' τους παρόχους υπηρεσιών νέφους |

☛ Κοινότητας (Community)

Το Νέφος Κοινότητας (Community Cloud) είναι παρεμφερές του Ιδιωτικού Νέφους, με τη διαφορά ότι η πρόσβαση επιτρέπεται σε περισσότερες από μία επιχειρήσεις, οι οποίες ως σύνολο ονομάζονται **Κοινότητα (community)**. Στεγάζεται σε μία ή περισσότερες επιχειρήσεις της κοινότητας, σε κάποιο εξωτερικό συνεργάτη ή σε συνδυασμό αυτών. Ιδανικό για οργανισμούς που επωφελούνται των προτερημάτων του public cloud αλλά χρειάζονται περισσότερα χαρακτηριστικά ασφάλειας.



Νέφος Κοινότητας
(Πηγή εικόνας:
<https://www.javatpoint.com/types-of-cloud>)

Παραδείγματα: cloud οργανισμού παροχής υπηρεσιών υγείας

Τα υπέρ και κατά του Community Cloud:

- | | |
|---|---|
| ✓ μικρότερη οικονομική επιβάρυνση ανά επιχείρηση (το κόστος κατανέμεται) | - χαμηλότερο επίπεδο ασφάλειας από το private cloud |
| ✓ υψηλότερο επίπεδο ασφάλειας | - κατάλληλο μόνο για οργανισμούς με συνεργάτες |
| ✓ παρέχει κατανομημένο περιβάλλον με δυνατότητα συνεργασίας | - διαμοιρασμός σταθερής χωρητικότητας και εύρους ζώνης σε όλα τα μέλη του community |
| ✓ κατανομή πόρων, υποδομών και άλλων δυνατοτήτων μεταξύ διαφόρων οργανισμών | |

2. Υπολογιστική Νέφος: Ορισμός, Αρχιτεκτονική και Υπηρεσίες

2.1 Cloud vs Cloud Storage vs Cloud Computing

Οι έννοιες **Νέφος (Cloud)**, **Αποθηκευτικός Χώρος Νέφους (Cloud Storage)** και **Υπολογιστική Νέφος (Cloud Computing)** συχνά συγχέονται, ωστόσο δεν έχουν ακριβώς την ίδια ερμηνεία.

Το **Νέφος**, όπως προαναφέρθηκε στην προηγούμενη υποενότητα, είναι ένα δίκτυο εξυπηρετητών το οποίο εξυπηρετεί απομακρυσμένους χρήστες αυτοματοποιώντας διαδικασίες.

Ο **Αποθηκευτικός Χώρος στο Νέφος** (ή **χωρητικότητα νέφους**), αφορά την αποθήκευση δεδομένων σε απομακρυσμένο εξυπηρετητή στο Νέφος, με στόχο την εξουσιοδοτημένη ανάκτησή τους μέσω του Διαδικτύου από πολλαπλές συσκευές.

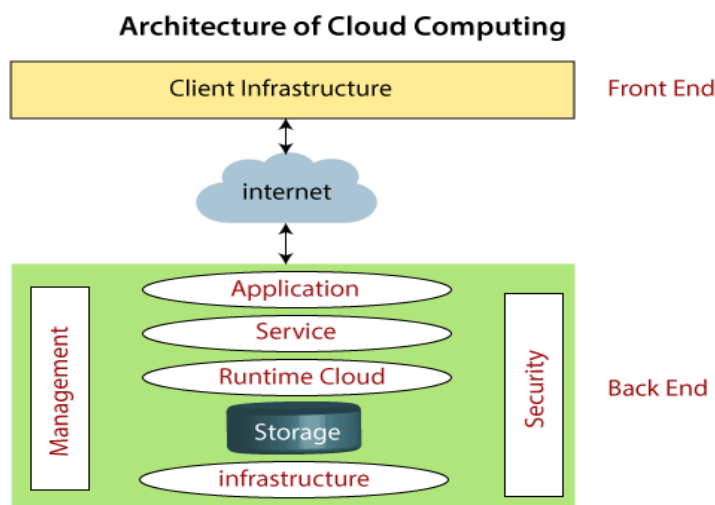
Τέλος, η **Υπολογιστική Νέφος**, ανήκει στο νέφος, και αφορά τη *διάθεση των υπολογιστικών πόρων του νέφους προκειμένου να είναι δυνατή η ανάλυση και τροποποίηση των δεδομένων που είναι αποθηκευμένα στον αποθηκευτικό χώρο του νέφους μέσω εκτέλεσης εφαρμογών.*

Επομένως, υπάρχει σύνδεση μεταξύ της Υπολογιστικής Νέφους και του Αποθηκευτικού Χώρου Νέφους.

Το υπολογιστικό νέφος, ως κομμάτι του νέφους, είναι και αυτό προσβάσιμο μόνο μέσω του Διαδικτύου.

2.2 Αρχιτεκτονική

Η αρχιτεκτονική του υπολογιστικού νέφους βασίζεται σε 2 κύρια συστατικά, το **front-end** και το **back-end**.



Αρχιτεκτονική Νέφους

(Πηγή εικόνας: <https://www.javatpoint.com/cloud-computing-architecture>)

Το **front-end** κομμάτι του νέφους, ή αλλιώς **Υποδομή Πελάτη (Client Infrastructure)**, αποτελεί όλες τις **διεπαφές** με τις οποίες οι χρήστες αλληλεπιδρούν άμεσα.

Οι συγκεκριμένες διεπαφές αποτελούν το διαμεσολαβητή μεταξύ του χρήστη και του αποθηκευτικού χώρου στο νέφος (cloud storage), και απαρτίζονται από **γραφική διεπαφή (GUI)** και **άλλα συστατικά υλικού και λογισμικού** (αποθήκευση δεδομένων (data storage), λογισμικό εικονικοποίησης (virtualization software) κτλ.).

Κάποια παραδείγματα της Υποδομής Πελάτη είναι οι Ελαφριοί Πελάτες (thin clients), όπως οι Φυλλομετρητές Ιστού (πχ. Chrome), οι κινητές συσκευές και οι ταμπλέτες.

Το **back-end** κομμάτι του νέφους χρησιμοποιείται από τον πάροχο υπηρεσιών νέφους, και **διευθύνει όλους τους υπολογιστικούς πόρους** που χρειάζονται για τις υπηρεσίες που προσφέρει ο πάροχος.

Αποτελείται από τεράστια χωρητικότητα, μηχανισμούς ασφαλείας, εικονικές μηχανές, μοντέλα ανάπτυξης, εξυπηρετητές και άλλα.

Όλα τα παραπάνω συστατικά του back-end του νέφους, μπορούν να κατηγοριοποιηθούν σε συνολικά 8 τύπους συστατικών, οι οποίοι είναι οι εξής:

☛ **Εφαρμογή (Application)**

Αφορά κάθε είδους λογισμικό ή πλατφόρμα στην οποία ένα χρήστης θέλει να έχει πρόσβαση.

☛ **Υπηρεσία (Service)**

Αφορά όλες τις υπηρεσίες που προσφέρει το νέφος. Το τμήμα αυτό διαχειρίζεται την πρόσβαση στις υπηρεσίες με βάση τον τύπο της υπηρεσίας που ζητά κάθε χρήστης. Υπάρχουν συνολικά 3 κατηγορίες/μοντέλα υπηρεσιών, οι οποίες θα αναλυθούν στην επόμενη υποενότητα.

☛ **Νέφος Χρόνου Εκτέλεσης (Runtime Cloud)**

Παρέχει το περιβάλλον εκτέλεσης στις εικονικές μηχανές.

☛ Αποθηκευτικός Χώρος (Storage)

Παρέχει ένα χώρο τεράστιας χωρητικότητας για αποθήκευση και διαχείριση δεδομένων. Αποτελεί ένα από τα πιο σημαντικά συστατικά της υπολογιστικής νέφους.

☛ Υποδομή (Infrastructure)

Παρέχει υπηρεσίες στα επίπεδα εφαρμογής και δικτύου. Αποτελείται από υλικό και λογισμικό απαραίτητα για την ομαλή λειτουργία του μοντέλου υπολογιστικής νέφους.

☛ Διαχείριση (Management)

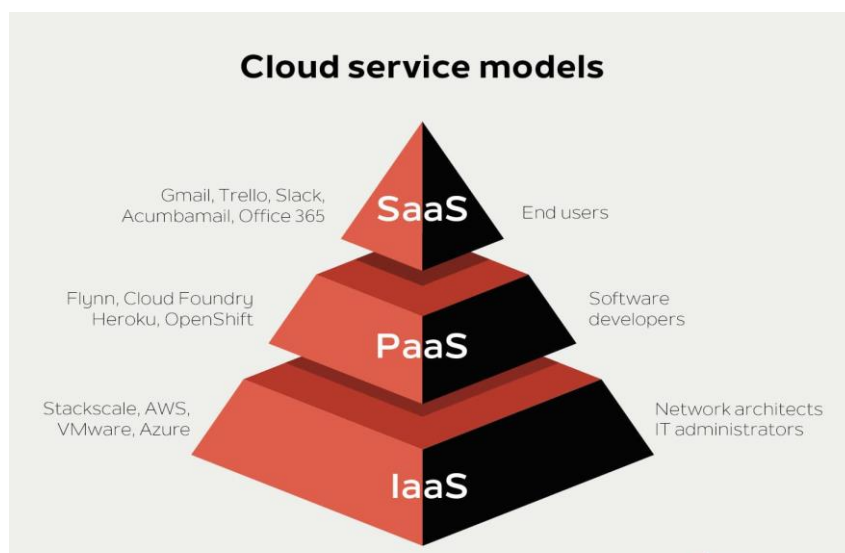
Διαχειρίζεται τα συστατικά του back-end και φροντίζει για τον συντονισμό τους.

☛ Ασφάλεια (Security)

Εφαρμόζει μηχανισμούς ασφαλείας στο back-end. Πρόκειται για ενσωματωμένο συστατικό του back-end της υπολογιστικής νέφους.

2.3 Μοντέλα υπηρεσιών νέφους

Οι υπηρεσίες που προσφέρονται από το νέφος χωρίζονται σε **3 μοντέλα υπηρεσιών (service models)**. Οι υπηρεσίες κατηγοριοποιούνται σε αυτά τα μοντέλα με βάση την αφαιρετικότητα τους.



Μοντέλα Υπηρεσιών Νέφους

(Πηγή εικόνας: <https://www.stackscale.com/blog/cloud-service-models/>)

☛ Υποδομή ως Υπηρεσία (Infrastructure as a Service - IaaS)

Γνωστό και ως **Υλικό ως Υπηρεσία (Hardware as a Service)**, πρόκειται για το πιο συνηθισμένο μοντέλο υπηρεσιών, καθώς παρέχει τη θεμελιώδη υποδομή για εικονικούς εξυπηρετητές, δίκτυα, λειτουργικά συστήματα και προγράμματα οδήγησης (drivers) αποθήκευσης δεδομένων.

Στο συγκεκριμένο μοντέλο, ο χρήστης χρησιμοποιεί **υπολογιστικούς πόρους του νέφους προσαρμόζοντας τα χαρακτηριστικά τους**, όπως η CPU, ανάλογα με τις απαιτήσεις που έχει. Φυσικά, δεν τροποποιούνται οι ίδιοι οι πόροι αλλά δημιουργούνται στιγμιότυπα αυτών ανά χρήση.

Κατά αυτό τον τρόπο ένας χρήστης μπορεί να αναπτύξει τις δικές του υπηρεσίες και συστήματα με χρήση αποκλειστικά απομακρυσμένου υλικού.

Η χρέωση γίνεται βάσει των απαιτήσεων του κάθε χρήστη (**pay-per-use**) και υποστηρίζεται σε public, private και hybrid υποδομές.

Παραδείγματα : AWS Elastic Compute Cloud (EC2)

Τα υπέρ και τα κατά της IaaS:

- | | |
|--|--|
| ✓ διαμοιραζόμενες υποδομές | - μέτριο επίπεδο |
| ✓ πρόσβαση στους πόρους μέσω Διαδικτύου | ασφάλειας |
| ✓ pay-per-use μοντέλο | - σε κάποιους οργανισμούς |
| ✓ εστίαση στην επιχείρηση και όχι στις υλικές υποδομές | δεν γίνεται αναβάθμιση του λογισμικού |
| ✓ κλιμακωσιμότητα κατ' απαίτηση | - μειωμένη δυνατότητα φορητότητας VM μεταξύ IaaS παρόχων |

☛ Πλατφόρμα ως Υπηρεσία (Platform as a Service - PaaS)

Στο συγκεκριμένο μοντέλο οι χρήστες **δεν έχουν τη δυνατότητα να προσαρμόσουν τις υλικές υποδομές** που θέλουν να χρησιμοποιήσουν, αλλά **έχουν πρόσβαση σε έτοιμες υποδομές**.

Παρέχεται δηλαδή **έτοιμο περιβάλλον** για δημιουργία, εκτέλεση ή έλεγχο λογισμικού. Το περιβάλλον αυτό απαρτίζεται από **γλώσσες προγραμματισμού, frameworks** εφαρμογών όπως Node.js και WordPress, **βάσεις δεδομένων** και άλλα εργαλεία, και συνήθως παρέχεται **με τη μορφή μιας εικονικής μηχανής**.

Όπως και στο IaaS μοντέλο, η χρέωση γίνεται βάσει του μοντέλου pay-per-use, και η πρόσβαση στις υπηρεσίες αυτές γίνεται μέσω Διαδικτύου.

Παραδείγματα : Google App Engine

Τα υπέρ και τα κατά της PaaS:

- | | |
|---|--|
| ✓ εστίαση στην ανάπτυξη λογισμικού και όχι στις υλικές υποδομές | - μειωμένη δυνατότητα φορητότητας εφαρμογών μεταξύ PaaS παρόχων |
| ✓ ελαχιστοποίηση οικονομικού ρίσκου (χρήση μόνο Η/Υ και Διαδικτύου) | - η ιδιωτικότητα των εταιρικών δεδομένων σε ρίσκο αν δεν βρίσκονται σε private cloud |
| ✓ παροχή έτοιμης λειτουργικότητας προς χρήση | - αυξημένη πολυπλοκότητα σε περίπτωση χρήσης τοπικών δεδομένων και δεδομένων στο cloud |
| ✓ κλιμακωσιμότητα | |

☛ **Λογισμικό ως Υπηρεσία (Software as a Service - SaaS)**

Γνωστό και ως **Κατ' απαίτηση Λογισμικό (On-Demand Software)**, πρόκειται για το μοντέλο με τη μεγαλύτερη αφαιρετικότητα όσο αφορά τις εφαρμογές και τα προϊόντα που παρέχει.

Οι προσφερόμενες υπηρεσίες είναι **προσβάσιμες από όλους μέσω Διαδικτύου**, επομένως δεν απαιτείται η εγκατάσταση κάποιου λογισμικού ή η χρήση κάποιας υλικής υποδομής, πέρα από τη συσκευή μέσω της οποίας ο χρήστης έχει πρόσβαση στο Διαδίκτυο με χρήση κάποιου προγράμματος περιήγησης.

Η χρέωση γίνεται μέσω συνδρομής ή βάσει του μοντέλου pay-per-use.

Παραδείγματα : Gmail, Google Docs, Photoshop

Τα υπέρ και τα κατά της SaaS:

- ✓ υποστήριξη μηνιαίας ή ετήσιας συνδρομής (μείωση κόστους)
- ✓ διαμοιρασμός στιγμιότυπου της εφαρμογής σε πολλούς χρήστες (one-to-many model)
- ✓ χρήση μόνο απομακρυσμένης υλικής υποδομής
- ✓ αυτόματες ενημερώσεις και διαδικασίες συντήρησης
- ✓ λειτουργικές εφαρμογές ανεξαρτήτου έκδοσης λογισμικού και υλικού πελάτη
- ✓ πρόσβαση μέσω οποιουδήποτε τύπου συσκευής
- ✓ εύκολη ενσωμάτωση άλλου λογισμικού ή υπηρεσιών μέσω APIs
- ✓ χωρίς εγκατάσταση λογισμικού
- μέτριο επίπεδο ασφάλειας
- αυξημένος χρόνος απόκρισης (latency) συγκριτικά με την τοπική εκτέλεση της εφαρμογής
- πλήρης εξάρτηση στο Διαδίκτυο
- μειωμένη δυνατότητα φορητότητας εφαρμογών μεταξύ SaaS παρόχων

3. Έλεγχοι Λογισμικού στο Νέφος: Ορισμός και Τύποι

3.1 Τι είναι ο έλεγχος λογισμικού

Ο **έλεγχος λογισμικού (software testing)**, είναι μία διαδικασία με στόχο την αξιολόγηση της λειτουργικότητας και της αξιοπιστίας ενός προγράμματος κατά την εκτέλεσή του. Η διαδικασία αυτή είναι πολύ σημαντικό κομμάτι του κύκλου ανάπτυξης ενός προϊόντος λογισμικού, καθώς μέσω αυτής εξασφαλίζεται ότι το προϊόν συμπεριφέρεται όπως πρέπει. Με τους ελέγχους λογισμικού εξασφαλίζονται η μείωση εξόδων ανάπτυξης, η βελτίωση της απόδοσης του προγράμματος, αλλά και η ελαχιστοποίηση των ευπαθειών του ως προς την ασφάλεια.

Ο έλεγχος λογισμικού μπορεί να γίνει με διάφορες στρατηγικές ως προς το τμήμα του προγράμματος που ελέγχεται κάθε φορά, τον αριθμό των χρηστών και την ροή των εισερχόμενων αιτημάτων, αλλά και την ευχρηστία του ως σύνολο.

3.2 Τι είναι Οι Έλεγχοι στο Νέφος

Οι έλεγχοι στο νέφος, αλλιώς **cloud testing**, είναι μία μορφή ελέγχων λογισμικού που χρησιμοποιούνται κυρίως για εφαρμογές στον Ιστό, και προσομοιώνουν τη λειτουργία των εφαρμογών αυτών στον πραγματικό κόσμο. Διεξάγονται από ειδικό ελεγκτή διασφάλισης ποιότητας σε περιβάλλον υπολογιστικού νέφους, με την βοήθεια παρεχόμενων υπηρεσιών του υπολογιστικού νέφους, οι οποίες προσαρμόζονται βάσει της ελεγχόμενης εφαρμογής. Εφαρμόζεται σε εξωτερικές εφαρμογές που μετατίθενται στο νέφος για έλεγχο, σε παρεχόμενες εφαρμογές του νέφους (SaaS), ή ακόμα και στην αρχιτεκτονική του νέφους (PaaS / IaaS).

Μία πρωτοποριακή υπηρεσία του υπολογιστικού νέφους είναι η **Testing as a Service**, κατά την οποία ο έλεγχος του λογισμικού δεν γίνεται από υπαλλήλους της επιχείρησης αλλά από εξωτερικό πάροχο υπηρεσιών. Η συγκεκριμένη υπηρεσία είναι ιδανική για επιχειρήσεις με έλλειψη ικανοτήτων ή πόρων επί του θέματος, αλλά όχι για αυτές που απαιτούν έλεγχο ειδικών υποδομών σε βάθος ή συγκεκριμένους τύπους ελέγχου.

3.3 Τύποι ελέγχου στο νέφος βάσει οντότητας – στόχου

Στη συγκεκριμένη υποενότητα αναλύονται διάφοροι τύποι ελέγχων λογισμικού, στόχος των οποίων είναι **να ελεγχθεί το ίδιο το νέφος** αλλά και οι εφαρμογές που παρέχει. Η κατηγοριοποίηση γίνεται με βάση το στόχο του ελέγχου.

☛ Έλεγχος Ολόκληρου του Νέφους (Testing the Whole Cloud)

Το νέφος αντιμετωπίζεται ως μία αυτόνομη οντότητα με συγκεκριμένα χαρακτηριστικά που προσφέρει. Οι έλεγχοι είναι σχεδιασμένοι με βάση αυτά τα δύο στοιχεία, και στοχεύουν στο νέφος ως οντότητα.

☛ Έλεγχος Στοιχείων εντός του Νέφους (Testing Elements within the cloud)

Το νέφος αντιμετωπίζεται ως ένα σύνολο στοιχείων, για κάθε ένα από τα οποία γίνεται έλεγχος. Τα στοιχεία αυτά αποτελούν τα εσωτερικά χαρακτηριστικά του νέφους.

☛ Έλεγχος Μεταξύ Νεφών (Testing across Clouds)

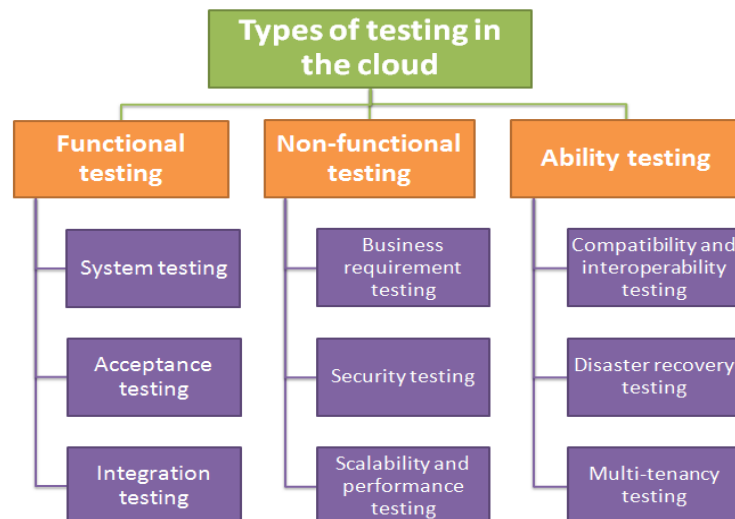
Οι έλεγχοι της συγκεκριμένης κατηγορίας γίνονται σε διάφορα μοντέλα ανάπτυξης νέφους, σύμφωνα με τις προδιαγραφές του κάθε ελέγχου.

☛ SaaS Έλεγχοι στο Νέφος (SaaS testing the cloud)

Η κατηγορία υπηρεσιών SaaS κάνει ελέγχους βάσει των απαιτήσεων του νέφους. Οι έλεγχοι αυτοί χωρίζονται σε λειτουργικούς και μη – λειτουργικούς, και αναλύονται στην επόμενη ενότητα.

4. Τεχνικές Ελέγχων στο Νέφος

Οι τεχνικές ελέγχου στο νέφος εστιάζουν στο τι πρέπει να εξασφαλιστεί βάσει ελέγχου και αφορούν όλους τους τύπους ελέγχου νέφους που αναπτύχθηκαν παραπάνω. Στόχος τους είναι η εξασφάλιση ότι η εφαρμογή που αναπτύχθηκε στο νέφος λειτουργεί όπως πρέπει, από άποψη λειτουργικότητας και προσδοκίας πελάτη.



Τεχνικές Ελέγχων στο Νέφος

(Πηγή εικόνας: <https://www.apriorit.com/dev-blog/548-cloud-based-testing>)

4.1 Λειτουργικός έλεγχος (Functional Testing)

Ο **λειτουργικός έλεγχος** ελέγχει αν το λογισμικό που αναπτύχθηκε καλύπτει όλες τις λειτουργικές απαιτήσεις που πρέπει.

Ο τρόπος με τον οποίο γίνεται ο έλεγχος είναι η **παροχή κατάλληλων δεδομένων εισόδου και η επικύρωση των δεδομένων εξόδου**, βάσει των λειτουργικών απαιτήσεων. Διεξάγεται στο περιβάλλον του υπολογιστικού νέφους, και αφορά και τοπικές και απομακρυσμένες εφαρμογές.

Ο έλεγχος γίνεται για κάθε μία λειτουργία του λογισμικού, και μπορεί να γίνει χειροκίνητα ή μέσω αυτοματοποίησης. Περιλαμβάνει κυρίως **έλεγχο μαύρου κουτιού (blackbox testing)**, δηλαδή έλεγχο του συστήματος με άγνοια της εσωτερικής δομής του λογισμικού, από άποψη κώδικα και άλλων εσωτερικών λεπτομερειών.

Λόγω της μορφής των λειτουργικών απαιτήσεων, ο συγκεκριμένος έλεγχος **δεν εστιάζει μόνο στη λειτουργικότητα του λογισμικού, αλλά και στην σύνδεση και επικοινωνία του με το υλικό**. Έτσι ελέγχονται διάφορες πτυχές του λογισμικού, όπως η ασφάλεια, η προσβασιμότητα καθώς και η συμπεριφορά του συστήματος σε καταστάσεις σφάλματος.

☛ Έλεγχος συστήματος (System Testing)

Ο έλεγχος συστήματος εστιάζει στην λειτουργικότητα των διαφόρων συναρτήσεων του συστήματος. Ελέγχει αν αυτή η λειτουργικότητα συμμορφώνεται με τις απαιτήσεις του συστήματος, και κατά πόσο η συμπεριφορά και ο σχεδιασμός του λογισμικού καλύπτουν τις προσδοκίες της επιχείρησης – πελάτη. Ονομάζεται και **Έλεγχος επαλήθευσης συστήματος**.

☛ Έλεγχος αποδοχής (Acceptance Testing)

Πρόκειται για τεχνική ελέγχου που πραγματοποιείται τόσο στο λογισμικό όσο και στους απομακρυσμένους εξυπηρετητές του νέφους. Κατά το συγκεκριμένο έλεγχο επαληθεύεται ότι η χρήση του νέφους πράγματι ικανοποιεί τις απαιτήσεις της επιχείρησης.

☛ Έλεγχος ενσωμάτωσης (Integration Testing)

Εξετάζεται η δυνατότητα ενσωμάτωσης του λογισμικού σε διαφορετικές πλατφόρμες και υποδομές του νέφους. Η συγκεκριμένη τεχνική είναι υπεύθυνη για τη σύνδεση του βασικού συστήματος με το σύστημα – στόχο, εξάγοντας δεδομένα απ' το βασικό σύστημα, αναγνωρίζοντας τη σημασιολογία και συντακτική δομή τους, και τελικά δημοσιεύοντας τα δεδομένα αυτά στο σύστημα – στόχο.

4.2 Μη λειτουργικός έλεγχος (Non-Functional Testing)

Γνωστός και ως **έλεγχος απόδοσης**, ο μη λειτουργικός έλεγχος είναι υπεύθυνος για τον **έλεγχο των μη λειτουργικών πτυχών του λογισμικού**, όπως η απόδοση, η ευχρηστία και η αξιοπιστία.

Η συγκεκριμένη τεχνική συμβάλλει στη **μείωση του ρίσκου και του κόστους παραγωγής**, καθώς βελτιστοποιεί το προϊόν όσο αφορά τις πτυχές που αναφέρθηκαν.

Επίσης συλλέγονται μετρήσεις που συνεισφέρουν στην αναβάθμιση του λογισμικού και βελτιώνουν τον τρόπο με τον οποίο γίνεται εγκατάσταση, ρύθμιση, εκτέλεση, διαχείριση και παρακολούθησή του.

Επομένως πρόκειται για μία τεχνική εξίσου σημαντική με τον λειτουργικό έλεγχο, καθώς επηρεάζει άμεσα τους τελικούς χρήστες.

☛ Έλεγχος απαιτήσεων επιχείρησης (Business Requirements Testing)

Οι εφαρμογές στο νέφος βασίζονται εξ ολοκλήρου στις απαιτήσεις της επιχείρησης, επομένως πριν την ανάπτυξη του λογισμικού στο νέφος ή την τοποθέτησή του σε αυτό, είναι απαραίτητη η εξέταση αυτών των απαιτήσεων. Αρχικά ορίζονται με μεγάλη ακρίβεια οι απαιτήσεις αυτές, μέσω συναντήσεων με τους πελάτες, εργασθηρίων και αξιολογήσεων.

Κομμάτι της συγκεκριμένης τεχνικής είναι και ο **έλεγχος διαθεσιμότητας νέφους**, το οποίο εξασφαλίζει ότι οι υπηρεσίες του νέφους είναι πάντα διαθέσιμες.

☛ Έλεγχος ασφάλειας (Security Testing)

Πρόκειται για έναν από τους πλέον πιο σημαντικούς ελέγχους, ρόλος του οποίου είναι η εξασφάλιση ότι τα δεδομένα της επιχείρησης αποθηκεύονται και μεταδίδονται με ασφαλή τρόπο.

Στο περιβάλλον του νέφους γίνεται ευρέα χρήση συσκευών ασφαλείας που εστιάζουν σε τρεις κύριες διαστάσεις: ακρίβεια, αποτελεσματικότητα και απόδοση.

☛ Έλεγχος κλιμακωσιμότητας και απόδοσης (Scalability and Performance Testing)

Ο έλεγχος κλιμακωσιμότητας και απόδοσης ανακαλύπτει τα όρια, τα σημεία συμφόρησης και τους περιορισμούς του συστήματος.

Συγκεκριμένα, γίνονται μετρήσεις των χρόνων απόκρισης και άλλων παραμέτρων που σχετίζονται με συγκεκριμένες ενέργειες, ενόσω το σύστημα δέχεται αιτήματα για διάφορες ενέργειες από πολλούς χρήστες. Ενδεικτικές παράμετροι που ελέγχονται είναι η χωρητικότητα, η ανταποκρισιμότητα, η αξιοπιστία και η κλιμακωσιμότητα.

Η συγκεκριμένη τεχνική ελέγχου χωρίζεται σε δύο επιμέρους τεχνικές, τον **Έλεγχο Φόρτου και Stress (Stress Testing)** και τον **Έλεγχο Χρόνου Απόκρισης**.

Η πρώτη τεχνική αφορά την ανταποκρισιμότητα και την αποτελεσματικότητα του συστήματος σε κατάσταση υψηλού φόρτου από πολλούς χρήστες.

Η δεύτερη τεχνική υπολογίζει την καθυστέρηση μεταξύ ενέργειας χρήστη και αντίστοιχης ανταπόκρισης του συστήματος, μετά την ανάπτυξη της εφαρμογής στο νέφος.

4.3 Έλεγχος Δυνατοτήτων (Ability Testing)

Ο έλεγχος δυνατοτήτων εξασφαλίζει ότι κάθε χρήστης λαμβάνει τις υπηρεσίες που απαιτεί απ' το νέφος σε πραγματικό χρόνο, και ορίζει τα όρια μέσα στα οποία αυτό είναι εφικτό.

Η συγκεκριμένη κατηγορία ελέγχων εντάσσεται από ένα μεγάλο ποσοστό στην κατηγορία μη λειτουργικού ελέγχου, ωστόσο για θέματα οργάνωσης αλλά και για να καλυφθεί η αντίληψη που δεν υποστηρίζει αυτή τη θέση έγινε διαχωρισμός αυτής απ' την προηγούμενη υποενότητα.

☛ Έλεγχος συμβατότητας και διαλειτουργικότητας (Compatibility and Interoperability Testing)

Κάθε εφαρμογή στο νέφος πρέπει να είναι λειτουργική σε πολλά περιβάλλοντα, και να μπορεί να εκτελεστεί σε ποικίλες πλατφόρμες του νέφους.

Ο συγκεκριμένος έλεγχος εξασφαλίζει αυτή τη δυνατότητα, διευκολύνοντας τη μετακίνηση μιας εφαρμογής νέφους απ' την μία υποδομή στην άλλη.

☛ Έλεγχος ανάκαμψης από καταστροφές (Disaster Recovery Testing)

Οι υπηρεσίες του νέφους πρέπει να είναι διαθέσιμες πάντα, ωστόσο πάντα υπάρχει η πιθανότητα μιας απρόβλεπτης ή αναπόφευκτης καταστροφής.

Σκοπός του ελέγχου ανάκαμψης από καταστροφές είναι οι επιπτώσεις της καταστροφής αυτής να περιοριστούν στο ελάχιστο, όπως ο χρόνος ανάκαμψης και η απώλεια δεδομένων.

☛ Έλεγχος Multi-Tenancy

Ο όρος multi-tenancy αναφέρεται στη χρήση ενός στιγμιotypu υπολογιστικού πόρου στο νέφος από πολλαπλές επιχειρήσεις – χρήστες.

Ο έλεγχος multi-tenancy είναι υπεύθυνος για την παροχή της δυνατότητας προσαρμογής του πόρου στις ανάγκες της κάθε επιχείρησης, αλλά και για τη διασφάλιση ότι κάθε επιχείρηση έχει πρόσβαση μόνο στα κομμάτια του πόρου που την αφορούν.

5. Τα οφέλη των Ελέγχων σε Περιβάλλον Νέφους

5.1 Η ανάγκη για γρήγορους ελέγχους λογισμικού στην σημερινή εποχή:

Στην σημερινή εποχή, ο χρόνος είναι σημαντικός παράγοντας για τις εταιρείες ανάπτυξης λογισμικού, εάν αυτές θέλουν να παραμένουν δυνατές απέναντι στον μεγάλο ανταγωνισμό που υπάρχει. Η **κυκλοφορία**, καθώς και η **ενημέρωση του λογισμικού**, είναι διαδικασίες που πρέπει να γίνονται κατά το δυνατόν ταχύτερα και ορθότερα.

Κάτι τέτοιο οδηγεί στην ανάγκη να εκτελείται με έναν αποδοτικότερο τρόπο και το στάδιο των **ελέγχων λογισμικού (software testing)** που είναι, όπως προαναφέρθηκε, ένα από τα πιο σημαντικότερα στάδια της ανάπτυξης λογισμικού, και απολύτως απαραίτητο για την διασφάλιση της ποιότητας και της σωστής λειτουργίας του λογισμικού.

Η εγκατάσταση ενός δοκιμαστικού **περιβάλλοντος προσομοίωσης στο νέφος** για την αυτοματοποίηση των ελέγχων λογισμικού είναι μια πρωτοπόρα μεθοδολογία σε σχέση με τις πιο παραδοσιακές τακτικές ελέγχου λογισμικού, , λόγω του ότι παρέχει πολυάριθμα πλεονεκτήματα σε θέματα που απασχολούν κάθε ομάδα ανάπτυξης λογισμικού.

5.2 Τα πλεονεκτήματα των υπηρεσιών νέφους για την αυτοματοποίηση των ελέγχων λογισμικού:

Παρακάτω, παρουσιάζονται αναλυτικά οι βασικότεροι από τους λόγους αυτούς, για τους οποίους η υπολογιστική νέφους φέρνει επανάσταση στην διαδικασία πραγματοποίησης ελέγχων λογισμικού

☛ Οι έλεγχοι στο νέφος είναι κατάλληλοι για πολλούς διαφορετικούς τύπους υλικού και λογισμικού.

Εφαρμογές που αναπτύσσονται σε περιβάλλον ιστού, στο νέφος, ή ακόμα και σε τοπικά μηχανήματα ενός οργανισμού, μπορούν να ελεγχθούν σε ένα δοκιμαστικό περιβάλλον νέφους, το οποίο προσομοιώνει την λειτουργία τους σε εικονικά μηχανήματα και πόρους λογισμικού.

☛ Οι έλεγχοι στο νέφος είναι εύκολα κλιμακώσιμοι.

Οι ελεγκτές λογισμικού μπορούν με ευκολία να αυξάνουν και να μειώνουν τους πόρους (μνήμη, χωρητικότητα, εικονικές μηχανές) που δεσμεύονται για την διαδικασία ελέγχου, ανάλογα με τις ανάγκες τους. Αυτό προσφέρει πολύ μεγάλη ευελιξία, που είναι κρίσιμη ιδιαίτερα σε περιπτώσεις που οι απαιτήσεις μιας εφαρμογής αλλάζουν συνεχώς.

☛ **Οι έλεγχοι στο νέφος έχουν χαμηλό χρηματικό κόστος.**

Στα περιβάλλοντα νέφους, οι χρεώσεις αντιστοιχούν μόνο στους πόρους που πραγματικά χρησιμοποιούνται. Έτσι, δεν απαιτείται η αγορά και συντήρηση ακριβού εξοπλισμού για την διενέργεια των ελέγχων. Η ομάδα ανάπτυξης λογισμικού πληρώνει μόνο το συγκεκριμένο δοκιμαστικό περιβάλλον που χρειάζεται για την εκάστοτε εφαρμογή. Συγκεκριμένα, μόνο τους πόρους που απαιτεί το περιβάλλον αυτό για την ορθή λειτουργία του και για να διεξαχθούν όλα τα απαιτούμενα συμπεράσματα για την ποιότητα του λογισμικού, και μόνο για το χρονικό διάστημα που πραγματικά θα χρησιμοποιηθούν, δηλαδή τις ώρες που θα διενεργούνται έλεγχοι.

☛ **Οι εικονικοί πόροι που δεσμεύονται στο νέφος μπορούν να αξιοποιούνται πολλές φορές την ίδια χρονική στιγμή.**

Κάτι τέτοιο δίνει πολύ μεγάλο πλεονέκτημα χρόνου και ευελιξία στην διαδικασία ελέγχου λογισμικού, αφού πολλοί προγραμματιστές μπορούν να διενεργούν τον έλεγχο ταυτόχρονα και από πολλές διαφορετικές γεωγραφικές τοποθεσίες. Αυτό είναι χρήσιμο, για διενέργεια ταυτόχρονων δοκιμών του λογισμικού σε περιβάλλοντα που παραμετροποιούνται με πολλούς διαφορετικούς τρόπους.

☛ **Τα περιβάλλοντα ελέγχων στο νέφος είναι εύκολα στην δημιουργία και διαμόρφωσή τους.**

Κάτι τέτοιο επιτρέπει στους ελεγκτές λογισμικού να προσομοιώνουν στο νέφος διαφορετικά περιβάλλοντα εκτέλεσης στα οποία θα «τρέχει» το λογισμικό στην πραγματική φάση λειτουργίας του, γρήγορα και χωρίς επιπλέον οικονομικό κόστος. Αυτό υποβοηθά την γρήγορη διενέργεια πολλών διαφορετικών τύπων ελέγχων, με πολλές παραλλαγές του συστήματος που προσομοιώνεται, όπως διαφορετικές πλατφόρμες, διαφορετικού τύπου υλικό, και άλλα.

☛ **Οι υπηρεσίες νέφους παρέχουν πολύ μεγάλη ποικιλία επιλογών για διαμόρφωση του περιβάλλοντος προσομοίωσης.**

Συγκεκριμένα, πολλά διαφορετικά λειτουργικά συστήματα, φυλλομετρητές, συσκευές, πλατφόρμες, κ.α. Έτσι, για παράδειγμα, μια ιστοσελίδα μπορεί να ελεγχθεί σε πολλούς διαφορετικούς φυλλομετρητές, μια εφαρμογή για κινητά σε διαφορετικές συσκευές, μια εφαρμογή desktop σε πολλές διαφορετικές εκδόσεις ενός λειτουργικού συστήματος κ.ο.κ.

☛ **Τα εργαλεία δημιουργίας περιβάλλοντος ελέγχου στο νέφος είναι συχνά εκ των προτέρων διαμορφωμένα από τον πάροχο τους.**

Αυτό κάνει την εκτέλεση ελέγχων σε διαφορετικά περιβάλλοντα προσομοίωσης μια διαδικασία ευκολότερη, γρηγορότερη, και λιγότερο επιρρεπή σε σφάλματα, απ' ό,τι με την χρήση παραδοσιακότερων τακτικών ελέγχου λογισμικού, στις οποίες οι ελεγκτές λογισμικού πρέπει να διαμορφώσουν εξ ολοκλήρου οι ίδιοι τα περιβάλλοντα ελέγχου.

☛ **Τα περιβάλλοντα νέφους είναι μονίμως σε διαθεσιμότητα.**

Τα περιβάλλοντα νέφους δεν παύουν ποτέ να βρίσκονται σε ετοιμότητα. Τα πραγματικά μηχανήματα των παρόχων υπηρεσιών νέφους λειτουργούν συνεχώς, και υπάρχουν πάντα επαρκείς πόροι υλικού και λογισμικού διαθέσιμοι. Αυτό είναι ένα σημαντικό πλεονέκτημα, καθώς δεν υπάρχει κανένας περιορισμός στην χρονική στιγμή και την διάρκεια που επιθυμεί μια ομάδα ανάπτυξης λογισμικού να πραγματοποιήσει τους ελέγχους.

☛ **Οι έλεγχοι στο νέφος βοηθούν την συνεργασία των προγραμματιστών σε μεγάλες ομάδες ανάπτυξης λογισμικού.**

Η συνεχής διαθεσιμότητά των υπηρεσιών νέφους, η δυνατότητα πρόσβασης σε αυτές από οποιοδήποτε μηχανήμα και οποιαδήποτε γεωγραφική τοποθεσία, καθώς επίσης και η δυνατότητα ταυτόχρονης χρήσης των πόρων, καθιστούν τα περιβάλλοντα ελέγχου στο νέφος την ιδανικότερη επιλογή για την πραγματοποίηση ελέγχων λογισμικού, όταν η ομάδα ανάπτυξης αποτελείται από πολλά άτομα.

☛ **Οι έλεγχοι σε ένα περιβάλλον νέφους εκτελούνται γρηγορότερα και πιο αξιόπιστα.**

Οι πάροχοι υπηρεσιών νέφους χρησιμοποιούν μεγάλο όγκο πραγματικών πόρων υλικού και λογισμικού για την δημιουργία εικονικών συσκευών, διακομιστών, containers εκτέλεσης λογισμικού, βάσεων δεδομένων, κ.α. Οι πόροι αυτοί λειτουργούν παράλληλα για την παροχή μιας γρήγορης και αξιόπιστης, με πολύ μικρή πιθανότητα σφαλμάτων, υπηρεσίας.

🛡️ **Η ανάκαμψη από σφάλματα είναι εύκολη στα περιβάλλοντα νέφους.**

Σε περίπτωση που συμβεί σφάλμα σε ένα περιβάλλον νέφους, ο χρόνος ανάκαμψης είναι χαμηλός. Λόγω του ότι προσομοιώνεται ένα περιβάλλον εκτέλεσης σε εικονικά μηχανήματα, σε περίπτωση βλάβης, το λογισμικό ή υλικό μπορεί να προσομοιωθεί εκ νέου πολύ γρήγορα σε κάποιο άλλο εικονικό μηχανήμα ή πόρο λογισμικού, και πιθανώς το σφάλμα δεν θα γίνει αντιληπτό από τους ελεγκτές λογισμικού.

Τα χαρακτηριστικά αυτά της υπολογιστικής νέφους, οδηγούν ολοένα και περισσότερες ομάδες ανάπτυξης λογισμικού, είτε αυτές ανήκουν σε μεγάλες εταιρείες, είτε σε μικρότερες, αυτοδημιούργητες επιχειρήσεις, να χρησιμοποιούν εργαλεία αυτοματοποίησης των ελέγχων λογισμικού με υπηρεσίες νέφους.

6. Τα Οφέλη του Νέφους για τους Εξ' Αποστάσεως Ελέγχους (Remote Testing)

6.1 Η ανάγκη για εξ' αποστάσεως πραγματοποίηση ελέγχων λογισμικού

Στις μέρες μας η εξ' αποστάσεως εργασία είναι αρκετά διαδεδομένη, καθώς με την συνεχή ανάπτυξη του Διαδικτύου και των μέσων κοινωνικής δικτύωσης, τα άτομα διαθέτουν τα μέσα για να επικοινωνούν με ευκολία μεταξύ τους, ανεξάρτητα της γεωγραφικής τοποθεσίας τους.



Εξ' αποστάσεως εργασία

(Πηγή εικόνας: <https://www.itpro.co.uk/business/business-strategy/356096/remote-working-are-you-ready-for-the-new-normal>)

Έτσι, τα τελευταία χρόνια ολοένα και περισσότεροι επαγγελματίες, ιδίως στον χώρο της Πληροφορικής, επιλέγουν μεθόδους εξ αποστάσεως εργασίας για την εύρεση συνεργατών και ευκαιριών επαγγελματικής εξέλιξης, σε όλο τον πλανήτη.

Λόγω της δύσκολης εποχής της πανδημίας του COVID-19, η εξ' αποστάσεως εργασία έχει γίνει για πολλούς ανάγκη, και όχι απλώς μια εναλλακτική μέθοδος εργασίας. Πολλοί επαγγελματίες δυσκολεύονται να προσαρμοστούν στα νέα δεδομένα, και οι ομάδες ανάπτυξης λογισμικού δεν αποτελούν εξαίρεση.

Παρά την μεγάλη εξοικείωση των προγραμματιστών με τους υπολογιστές, η εξ' αποστάσεως εργασία πάνω σε ένα έργο λογισμικού απαιτεί πολύ καλύτερες ικανότητες συνεργασίας, συνεννόησης και χρονοπρογραμματισμού έργου, από όλα τα μέλη της ομάδας, απ' ότι η διαζώσης υλοποίηση του ίδιου έργου.

Αυτό φυσικά ισχύει και για τους ελέγχους λογισμικού, οι οποίοι αποτελούν ένα πολύ σημαντικό στάδιο της ανάπτυξης έργων λογισμικού και απαιτεί πολύ καλή συνεργασία των μελών της ομάδας που το αναλαμβάνει.

6.2 Προβλήματα της εξ' αποστάσεως πραγματοποίησης ελέγχων

Με την εξ' αποστάσεως εργασία προκύπτουν αρκετά ζητήματα οργάνωσης και απόδοσης των μελών μιας ομάδας που εργάζεται πάνω στην διεξαγωγή των ελέγχων ενός έργου λογισμικού.

Οι έλεγχοι στο νέφος, λόγω των διαφόρων πλεονεκτημάτων που προσφέρουν, μπορούν να αποβούν πολύ πιο αποδοτικοί για τους εξ' αποστάσεως ελέγχους απέναντι σε άλλες τακτικές ελέγχων λογισμικού.

Στην παρούσα ενότητα αναλύονται οι **βασικότερες προκλήσεις** για τους εξ' αποστάσεως ελέγχους λογισμικού, και **πως οι υπηρεσίες υπολογιστικής νέφους φέρνουν λύσεις** στα ζητήματα αυτά.

Η γεωγραφική απόσταση των μελών της ομάδας ελέγχου από τους πραγματικούς πόρους υλικού και λογισμικού.

Το γεγονός ότι τα άτομα που εργάζονται πάνω στους ελέγχους ενός έργου λογισμικού δεν βρίσκονται στην ίδια τοποθεσία την στιγμή που εργάζονται, σημαίνει πως αυτά δεν έχουν την δυνατότητα να χρησιμοποιούν τους ίδιους πραγματικούς πόρους, όπως μηχανήματα.

Η εικονικοποίηση ενός περιβάλλοντος ελέγχου είναι δύσκολο και χρονοβόρο να πραγματοποιηθεί από κάθε μέλος της ομάδας ξεχωριστά.

Κάτι τέτοιο μπορεί να αποβεί πολύ κοστοβόρο, εάν κάθε μέλος της ομάδας αναγκαστεί να προμηθευτεί ξεχωριστά συστατικά για την πραγματοποίηση των ελέγχων. Για τα περισσότερα έργα λογισμικού, ένα τέτοιο κόστος είναι απαγορευτικό.

Πως βοηθά το νέφος: Οι υπηρεσίες νέφους, κάνουν εύκολη την δημιουργία ενός περιβάλλοντος προσομοίωσης για την αυτοματοποίηση των ελέγχων με δυνατότητα επιλογής από πολυάριθμους εικονικούς πόρους υλικού και λογισμικού, στους οποίους επιτρέπεται η παράλληλη πρόσβαση από πολλές διαφορετικές τοποθεσίες.

Έτσι, τα μέλη της ομάδας μπορούν να μοιράζονται τους πόρους τους ακόμα και εάν δεν βρίσκονται στην ίδια γεωγραφική τοποθεσία με αυτούς.

☛ **Δυσκολίες στην επικοινωνία των μελών της ομάδας και η ανάγκη για αποδοτική συνεργασία ατόμων σε διαφορετικές ζώνες ώρας.**

Όταν οι έλεγχοι λογισμικού διενεργούνται εξ' αποστάσεως, τα μέλη της ομάδας πρέπει να έχουν την δυνατότητα να επικοινωνούν συνεχώς όσο εργάζονται, ώστε να καταφέρνουν να οργανώνουν και να συγχρονίζουν τις ενέργειες τους.

Η χρήση μεθόδων όπως οι βιντεοκλήσεις και το screen share, μπορούν να αποτελέσουν λύσεις σε αυτό το ζήτημα όμως είναι αρκετά προβληματικές λόγω τεχνικών προβλημάτων, κακής σύνδεσης στο Διαδίκτυο, ή δυσκολία οργάνωσης συναντήσεων λόγω διαφορετικών ζωνών ώρας.

Αυτό φέρνει την ανάγκη να μπορεί κάθε μέλος της ομάδας να εργάζεται, χωρίς να πρέπει ταυτοχρόνως να επικοινωνεί προφορικά με τους συνεργάτες του.

Πως βοηθά το νέφος: Το γεγονός ότι οι υπηρεσίες νέφους είναι συνεχώς διαθέσιμες (up-and-running) βοηθά την εξ' αποστάσεως συνεργασία μιας ομάδας, ιδίως όταν τα μέλη της βρίσκονται σε τοποθεσίες με διαφορετικές ζώνες ώρας.

Ο κάθε προγραμματιστής μπορεί με ευκολία να κάνει το μέρος της δουλειάς που του αντιστοιχεί, χωρίς να απαιτείται η διοργάνωση μιας συνάντησης με άλλα μέλη της ομάδας.

Έτσι, αρκεί μια σωστή ανάθεση αρμοδιοτήτων στα μέλη της ομάδας, ώστε να ολοκληρωθεί το έργο αποδοτικά.

7. Οι Δυσκολίες των Ελέγχων σε Περιβάλλον Νέφους

7.1 Οι προκλήσεις των ελέγχων λογισμικού στο νέφος

Παρά τα πολυάριθμα πλεονεκτήματα της υπολογιστικής νέφους για τους ελέγχους λογισμικού, υπάρχουν ορισμένες προκλήσεις για τις ομάδες που επιλέγουν να πραγματοποιήσουν στο νέφος αυτό το στάδιο της ανάπτυξης λογισμικού.

Όλες οι ομάδες πρέπει να λαμβάνουν υπόψιν τους ενδεχόμενους κινδύνους για την πορεία του έργου που μπορεί να επιφέρει η μετάβαση σε υπηρεσίες νέφους, προτού λάβουν την απόφαση να ξεκινήσουν τη διαδικασία των ελέγχων τους εκεί.

Η πραγματοποίηση ελέγχων στο νέφος δημιουργεί κινδύνους ασφάλειας.

Παρά το γεγονός ότι στο νέφος χρησιμοποιείται κρυπτογράφηση, καθώς επίσης δίνονται ορισμένες εγγυήσεις ασφάλειας από τους παρόχους των υπηρεσιών αυτών, υπάρχει πάντα ένα ενδεχόμενο παραβίασης της ασφάλειας των δεδομένων. Ακόμα κι αν χρησιμοποιείται ένα private cloud, τα δεδομένα εξακολουθούν να αποθηκεύονται σε απομακρυσμένους διακομιστές, την πραγματική τοποθεσία των οποίων δεν γνωρίζει κανένα μέλος της ομάδας ανάπτυξης λογισμικού. Ιδιαίτερα όταν μια εφαρμογή διαχειρίζεται ευαίσθητα δεδομένα, είναι προτιμότερο τα δεδομένα αυτά να αποθηκεύονται σε τοπικά μηχανήματα του οργανισμού του οποίου το λογισμικό είναι ιδιοκτησία.

Δεν προσφέρουν όλοι οι πάροχοι τις ίδιες υπηρεσίες και την ίδια ποικιλία πόρων.

Αυτό μπορεί να οδηγήσει σε δυσκολία εύρεσης ενός παρόχου νέφους που να προσφέρει όλες τις υπηρεσίες τις οποίες χρειάζονται για την προσομοίωση όλων των πιθανών σεναρίων χρήσης του λογισμικού. Η πραγματοποίηση των ελέγχων σε πολλούς διαφορετικούς παρόχους ταυτόχρονα κάνει την διαδικασία πολύ περίπλοκη και μη διαχειρίσιμη.

☛ **Ενδεχόμενα προβλήματα διαθεσιμότητας των παρόχων υπηρεσιών νέφους.**

Παρόλο που οι πάροχοι νέφους εγγυούνται 24 ώρες το 24ωρο διαθεσιμότητα των υπηρεσιών τους, έχουν υπάρξει περιπτώσεις όπου ακόμα και οι μεγαλύτεροι πάροχοι αντιμετώπισαν κάποια λειτουργικά προβλήματα για μικρά χρονικά διαστήματα. Ωστόσο, ακόμα και μικρό διάστημα μη διαθεσιμότητας μπορεί να αποβεί μοιραίο για τους ελέγχους κρίσιμου λογισμικού, όποτε σε τέτοιες περιπτώσεις είναι καλύτερο να επιλέγεται για τους ελέγχους μια μέθοδος που να εξασφαλίζει 100% διαθεσιμότητα, κάτι που δεν μπορεί να ισχύει στην περίπτωση ενός απομακρυσμένου διακομιστή, αγνώστου τοποθεσίας, και προσβάσιμου μόνο μέσω Διαδικτύου, όπως αυτοί του νέφους.

☛ **«Κρυμμένο» χρηματικό κόστος.**

Παρόλο που οι υπηρεσίες νέφους παρέχουν εικονικούς πόρους σε τιμές πολύ χαμηλότερες από την αγορά πραγματικών πόρων για την πραγματοποίηση των ελέγχων, η πολιτική κοστολόγησης «pay as you go» («πλήρωσε μόνο για ότι χρησιμοποιείς»), μπορεί να επιφέρει μεγάλη χρηματική ζημιά αν οι έλεγχοι δεν είναι εκ των προτέρων πολύ σωστά οργανωμένοι. Η ομάδα ελέγχου λογισμικού πρέπει να είναι σωστά ενημερωμένη για τον τρόπο λειτουργίας και κοστολόγησης της κάθε συγκεκριμένης υπηρεσίας η οποία θα χρησιμοποιηθεί, ώστε να μην προκύψουν λάθη που θα αποβούν πολύ κοστοβόρα για το έργο.

☛ **Η διαχείριση ενός περιβάλλοντος ελέγχου στο νέφος δεν είναι πάντα εύκολη.**

Εάν η ομάδα ανάπτυξης λογισμικού δεν είναι αρκετά εξοικειωμένη με τις υπηρεσίες νέφους, η εγκατάσταση και διαχείριση ενός περιβάλλοντος ελέγχου στο νέφος μπορεί να είναι πολύ περίπλοκη διαδικασία και να οδηγήσει σε σφάλματα που να επιφέρουν κόστος χρημάτων ή και χρόνου, που να αποβεί μοιραίο για την πορεία του έργου.

☛ **Το νέφος μπορεί να μειώσει την ταχύτητα εκτέλεσης των ελέγχων.**

Το γεγονός ότι οι υπηρεσίες νέφους είναι προσβάσιμες από πολλούς χρήστες σε πολλές τοποθεσίες ταυτόχρονα, σε συνδυασμό με το ότι φιλοξενούνται σε απομακρυσμένους, κατανεμημένους διακομιστές, μπορεί να οδηγήσει σε μεγάλες καθυστερήσεις στην εκτέλεση των ελέγχων.

☁ **Το νέφος δεν είναι πάντα η φθηνότερη επιλογή.**

Η πολιτική κοστολόγησης «pay as you go» οδηγεί στο να μην είναι τελικά οι υπηρεσίες νέφους η φθηνότερη επιλογή για ορισμένες κατηγορίες λογισμικού. Ιδιαίτερα για το λογισμικό αυτό που οι απαιτήσεις του αλλάζουν συνεχώς ή για εφαρμογές που κλιμακώνονται ραγδαία. Όσο περισσότεροι πόροι χρησιμοποιούνται, τόσο ακριβότερη είναι η υπηρεσία. Γι' αυτό η ομάδα ανάπτυξης λογισμικού πρέπει να υπολογίζει εκ των προτέρων την ποσότητα των πόρων που απαιτείται για τους ελέγχους, και να κρίνει εάν οι υπηρεσίες νέφους είναι τελικά πιο συμφέρουσες από την αγορά πραγματικών πόρων.

☁ **Η μετάβαση από έναν πάροχο νέφους σε έναν άλλο δεν είναι εύκολη.**

Ένας πάροχος νέφους μπορεί να είναι αρχικά ιδανικός για της ανάγκες ενός έργου λογισμικού. Ωστόσο, οι ανάγκες των εφαρμογών αλλάζουν όσο αυτές κλιμακώνονται ή εάν τροποποιούνται οι επιχειρησιακές τους απαιτήσεις. Σε τέτοιες περιπτώσεις, μπορεί ο επιλεγμένος πάροχος νέφους να πάψει να καλύπτει της ανάγκες σε πόρους για τους ελέγχους λογισμικού ή το κόστος να είναι πολύ μεγάλο για την κάλυψη των αναγκών αυτών. Τότε, η μετάβαση σε έναν νέο πάροχο νέφους είναι δύσκολη και κοστοβόρα διαδικασία, αφού απαιτεί την δημιουργία του περιβάλλοντος ελέγχου εξ ολοκλήρου από την αρχή. Αυτό είναι ένα χαρακτηριστικό του νέφους που το καθιστά ακατάλληλο για την πραγματοποίηση των ελέγχων εφαρμογών οι οποίες κλιμακώνονται ραγδαία ή αλλάζουν συνεχώς οι απαιτήσεις τους.

☁ **Η παρακολούθηση της απόδοσης των ελέγχων στο νέφος μπορεί να είναι περίπλοκη.**

Ιδιαίτερα όταν η αρχιτεκτονική της εφαρμογής είναι περίπλοκη, η παρακολούθηση της απόδοσης των ελέγχων, που είναι πολύ σημαντική για την πορεία ενός έργου ανάπτυξης λογισμικού, είναι δυσκολότερη στο νέφος, ιδίως εάν οι προγραμματιστές της ομάδας δεν κατέχουν την απαραίτητη εξοικείωση με τις υπηρεσίες νέφους.

7.2 Πότε το νέφος δεν είναι η κατάλληλη επιλογή

Με βάση τα παραπάνω μειονεκτήματα των υπηρεσιών νέφους, κρίνεται πως οι έλεγχοι στο νέφος δεν είναι η καταλληλότερη επιλογή στις ακόλουθες περιπτώσεις:

- ☛ Όταν η εφαρμογή που αναπτύσσεται επεξεργάζεται ευαίσθητα δεδομένα όπως δεδομένα πληρωμών, τοποθεσίες, ευαίσθητα προσωπικά δεδομένα κ.α.
- ☛ Σε περιπτώσεις που η εφαρμογή πραγματοποιεί κρίσιμες λειτουργίες οι οποίες βασίζονται στην συνεχή διαθεσιμότητα των δεδομένων της, καθώς η μη διαθεσιμότητα μπορεί να αποβεί καταστροφική και πολύ κοστοβόρα.
- ☛ Σε περιπτώσεις που η εφαρμογή κλιμακώνεται ραγδαία και απαιτεί συνεχείς μετακινήσεις από υπηρεσία σε υπηρεσία και από πάροχο σε πάροχο.
- ☛ Εάν η αρχιτεκτονική της εφαρμογής είναι αρκετά περίπλοκη και αποτελείται από πολλά διαφορετικά συστατικά, ιδίως αν τα συστατικά αυτά δεν είναι διαθέσιμα από μονάχα έναν κοινό πάροχο υπηρεσιών νέφους.
- ☛ Σε περιπτώσεις εφαρμογών πολύ μεγάλης κλίμακας, όπου οι ομάδες ανάπτυξής τους έχουν την οικονομική δυνατότητα να τις φιλοξενήσουν με δικούς τους πόρους υλικού και λογισμικού.
- ☛ Όταν η ομάδα ανάπτυξης λογισμικού δεν έχει την δυνατότητα να παρακολουθεί συνεχώς την απόδοση των ελέγχων της.
- ☛ Για εφαρμογές “legacy”, δηλαδή παλιές εκδόσεις εφαρμογών για συμβατότητα με παλαιότερες τεχνολογίες υλικού και λογισμικού, οι οποίες απαιτούν ήδη πολύ μεγάλο χρηματικό κόστος για την συντήρησή τους.

8. Η Σωστή Κατεύθυνση Προετοιμασίας Μετάβασης στο Νέφος για Ελέγχους

8.1 Η ανάγκη για σωστή οργάνωση των έργων λογισμικού

Σε οποιοδήποτε έργο λογισμικού, από μεγάλα συστήματα με δεκάδες χιλιάδες γραμμές κώδικα που αναπτύσσονται από μεγάλες ομάδες, μέχρι και απλούστερες εφαρμογές όπως μια ιστοσελίδα ή μια εφαρμογή για κινητά, η σωστή προετοιμασία και η μεθοδικότητα οδηγούν σε ένα καλύτερο αποτέλεσμα.

Η έλλειψη οργάνωσης επιφέρει αύξηση των πιθανοτήτων εμφάνισης σφαλμάτων που ίσως αποβούν μοιραία για την πορεία ανάπτυξης ενός έργου λογισμικού, αφού μπορεί να οδηγήσουν σε οικονομική ζημία, αποκλίσεις από τον χρονοπρογραμματισμό των εργασιών, μέχρι και αποτυχία ολοκλήρωσης του έργου.

Έτσι, και για την σωστή πραγματοποίηση των ελέγχων λογισμικού σε ένα περιβάλλον νέφους, χρειάζεται η ομάδα ανάπτυξης να προετοιμαστεί κατάλληλα.



Μετάβαση στο Νέφος

(Πηγή Εικόνας: <https://www.breeze.pm/blog/5-reasons-why-moving-your-business-to-the-cloud-is-a-good-idea>)

Είναι σημαντικό, προτού ξεκινήσει την διαδικασία μετάβασης στο νέφος για την πραγματοποίηση των ελέγχων, η ομάδα να βεβαιωθεί **πως η υπολογιστική νέφος είναι η κατάλληλη επιλογή** για το εκάστοτε έργο.

Κατά την απόφαση αυτή, πρέπει να λαμβάνονται υπόψιν τόσο τα θετικά **όσο και οι δυσκολίες που μπορεί να προκύψουν** με την χρήση των υπηρεσιών νέφους, όπως αυτές περιεγράφηκαν στην προηγούμενη ενότητα.

Η επιλογή πρέπει να γίνεται συνειδητά, με βάση τις ανάγκες, την κλίμακα και τις απαιτήσεις της εφαρμογής, αλλά και την σύσταση της ομάδας ανάπτυξης, τον αριθμό των ατόμων και την εξοικείωση τους με τις υπηρεσίες νέφους.

8.2 Βήματα για την σωστή προετοιμασία αυτοματοποίησης των ελέγχων με υπηρεσίες νέφους

Εφόσον μια ομάδα επιλέξει να πραγματοποιήσει τους ελέγχους της με χρήση υπηρεσιών υπολογιστικής νέφους, πρέπει να προετοιμαστεί κατάλληλα. Παρακάτω δίνεται μια περιγραφή ορισμένων χρήσιμων τακτικών και βημάτων για την προετοιμασία αυτή:

☛ Σχεδιασμός μιας ολοκληρωμένης στρατηγικής για την πραγματοποίηση των ελέγχων.

Πιο συγκεκριμένα, η ομάδα ανάπτυξης θα πρέπει εκ των προτέρων να ορίσει ξεκάθαρα το είδος των ελέγχων, καθώς και με τι μέσα θα πραγματοποιηθούν αυτοί, προτού δημιουργήσει ένα περιβάλλον προσομοίωσης στο νέφος για την κάλυψη όλων των αναγκών των ελέγχων. Η έλλειψη μιας τέτοιας στρατηγικής μπορεί να οδηγήσει σε απώλεια χρόνου ή χρημάτων λόγω σφαλμάτων στους ελέγχους.

☛ Σχεδιασμός της υποδομής του περιβάλλοντος πραγματοποίησης ελέγχων λογισμικού.

Κατά τον σχεδιασμό της στρατηγικής των ελέγχων, είναι σημαντικό η ομάδα ανάπτυξης λογισμικού να ορίσει επίσης τις ακριβείς απαιτήσεις της αρχιτεκτονικής του περιβάλλοντος ελέγχου που θέλει να προσομοιώσει. Έτσι, θα μπορεί ύστερα να επιλέξει έναν κατάλληλο πάροχο νέφους με βάση αυτές τις απαιτήσεις που πρέπει να καλυφθούν.

☛ Έρευνα αγοράς και επιλογή του κατάλληλου πάροχου υπηρεσιών νέφους

Έχοντας έτοιμη μια στρατηγική ελέγχων και ένα σχέδιο της υποδομής του περιβάλλοντος προσομοίωσης, η ομάδα ανάπτυξης λογισμικού μπορεί πλέον να προχωρήσει σε έρευνα αγοράς και επιλογή του κατάλληλου παρόχου νέφους. Ο πάροχος αυτός θα πρέπει να προσφέρει όλους τους απαραίτητους τύπους πόρων. Εάν υπάρχουν παραπάνω από ένας πάροχοι με τις αναγκαίες υπηρεσίες, η έρευνα αγοράς βοηθά στην επιλογή εκείνου που συμφέρει περισσότερο από οικονομική άποψη.

☛ **Δοκιμαστική λειτουργία σε δωρεάν υπηρεσίες νέφους (free trials)**

Οι περισσότεροι πάροχοι νέφους δίνουν την δυνατότητα μιας δοκιμαστικής περιόδου χρήσης των υπηρεσιών τους. Κάτι τέτοιο, προσφέρει στις ομάδες ανάπτυξης λογισμικού τη δυνατότητα να δοκιμάσουν τις υπηρεσίες αυτές, να δημιουργήσουν δοκιμαστικά στιγμιότυπα πόρων και να πάρουν μια σωστότερη απόφαση για το εάν ο συγκεκριμένος πάροχος νέφους πράγματι καλύπτει τις ανάγκες των ελέγχων του εκάστοτε έργου.

☛ **Ανάθεση αρμοδιοτήτων και δικαιωμάτων πρόσβασης στις υπηρεσίες νέφους για όλα τα μέλη της ομάδας ελέγχου λογισμικού**

Είναι σημαντικό επίσης να γίνει μια προσεκτική ανάθεση αρμοδιοτήτων ανάμεσα στα μέλη της ομάδας που θα αναλάβουν τους ελέγχους λογισμικού. Πρέπει η διαδικασία ελέγχου να οργανωθεί προσεκτικά ως προς το με ποια σειρά θα γίνουν όλοι οι απαραίτητοι έλεγχοι και από ποιους. Ύστερα, για κάθε επιμέρους έλεγχο, να ανατεθούν αρμοδιότητες στον κάθε προγραμματιστή και να του δοθούν τα αντίστοιχα δικαιώματα πρόσβασης στους αναγκαίους πόρους νέφους. Έτσι, θα είναι ευκολότερη η παρακολούθηση της πορείας των ελέγχων καθώς και των αποτελεσμάτων αυτών.

☛ **Συνεχής παρακολούθηση και ανάλυση των αποτελεσμάτων των ελέγχων**

Είναι απαραίτητο, κατά την διάρκεια της διεξαγωγής των ελέγχων να γίνεται συνεχής παρακολούθηση της πορείας και των αποτελεσμάτων τους. Γι' αυτό τον λόγο είναι σημαντικό κάποια από τα μέλη της ομάδας να έχουν την απαραίτητη εξοικείωση με το νέφος. Ακόμα, υπάρχουν ειδικοί έλεγχοι λογισμικού στο νέφος, οι οποίοι εξειδικεύονται αποκλειστικά στο συγκεκριμένο είδος ελέγχων. Επίσης, μπορεί να γίνεται χρήση κατάλληλων εργαλείων.

9. Εργαλεία Ελέγχου στο Νέφος

Η συνεχώς αυξανόμενη δημοφιλία της χρήσης του νέφους για την πραγματοποίηση του σταδίου ελέγχων λογισμικού έχει οδηγήσει στην ανάπτυξη μεγάλης ποικιλίας εργαλείων βασισμένα σε υπηρεσίες νέφους. Η χρήση τέτοιων εργαλείων βοηθά στην ευκολότερη διεξαγωγή των ελέγχων.

- ☁ Παρέχουν έτοιμα σύνολα εντολών και εκ των προτέρων ρυθμισμένα περιβάλλοντα ελέγχων, κάνοντας ευκολότερη την κατασκευή της υποδομής ελέγχων βασισμένη σε υπηρεσίες νέφους και αυτοματοποιώντας σε μεγάλο βαθμό την διεξαγωγή των ελέγχων.
- ☁ Διευκολύνουν την παρακολούθηση των αποτελεσμάτων μέσω εύχρηστων διεπαφών.
- ☁ Γενικά, παρέχουν πλεονέκτημα σε προγραμματιστές με λιγότερη εξοικείωση με τις υπηρεσίες νέφους αρά και στην συνεργασία μεγάλων ομάδων ανάπτυξης.

9.1 Δημοφιλή εργαλεία ελέγχου στο νέφος

Τα ακόλουθα είναι μερικά από τα πιο ευρέως χρησιμοποιούμενα εργαλεία ελέγχων βασισμένων σε υπηρεσίες νέφους που κυκλοφορούν στην αγορά σήμερα:

- ☁ **SOASTA CloudTest**
Εργαλείο για Λειτουργικούς Ελέγχους και Ελέγχους Απόδοσης σε εφαρμογές εγκατεστημένες σε πλατφόρμες ιστού και κινητές εφαρμογές. Προσομοιώνει την λειτουργία χιλιάδων χρηστών ταυτόχρονα για έλεγχο των εφαρμογών σε συνθήκες πολύ μεγάλου φόρτου επεξεργασίας. Παρέχει εργαλεία ανάλυσης των αποτελεσμάτων σε πραγματικό χρόνο με δυνατότητα αυξομείωσης του φόρτου κατά την διάρκεια του ελέγχου. Χρησιμοποιεί δημοφιλείς παρόχους νέφους όπως ο Amazon AWS.
- ☁ **LoadStorm**
Ένα ακόμα εργαλείο για Ελέγχους Απόδοσης σε εφαρμογές ιστού και κινητές εφαρμογές. Ένα εύκολα κλιμακώσιμο εργαλείο, ιδανικό για εύρεση του ορίου χρηστών που μπορεί να εξυπηρετεί ταυτόχρονα η εφαρμογή.
- ☁ **AppPerfect**
Εργαλείο ελέγχου Ασφάλειας και ελέγχου Απόδοσης σε εφαρμογές ιστού. Δίνει επίσης την δυνατότητα ελέγχου Διαλειτουργικότητας (Interoperability) καθώς παρέχει ποικιλία φυλλομετρητών, λειτουργικών συστημάτων και συσκευών για διεξαγωγή ελέγχων.

☛ **Xamarin Test Cloud**

Εργαλείο Ελέγχου Αποδοχής (Acceptance Testing) για γραφικές διεπαφές εφαρμογών σε κινητές συσκευές.

☛ **Nessus**

Εργαλείο για Ελέγχους Δεισδυσσης (Penetration Testing). Παρέχει μια μεγάλη βιβλιοθήκη ευπαθειών για εφαρμογές νέφους. Είναι συμβατό με μεγάλους παρόχους νέφους όπως Amazon AWS, Azure, Google Cloud.

Φυσικά, υπάρχουν περισσότερα εργαλεία ελέγχων βασισμένα σε υπηρεσίες νέφους, που ικανοποιούν διαφορετικές ανάγκες ελέγχων για διαφορετικούς τύπους εφαρμογών.

10. Σύνοψη

Το νέφος είναι κατάλληλο για την διεξαγωγή πολλών διαφορετικών τύπων ελέγχου λογισμικού, για διάφορους τύπους εφαρμογών, με την προσομοίωση ενός περιβάλλοντος ελέγχου με χρήση των εικονικών πόρων και υπηρεσιών της υπολογιστικής νέφους.

Οι έλεγχοι στο νέφος μπορούν να προσφέρουν πληθώρα πλεονεκτημάτων για τις ομάδες ανάπτυξης λογισμικού, όμως υπάρχουν και ορισμένα σημαντικά μειονεκτήματα και προκλήσεις. Αυτά, πρέπει πάντοτε να λαμβάνονται υπόψιν από τους ελεγκτές λογισμικού, πριν επιλέξουν το νέφος ως την κύρια πλατφόρμα διεξαγωγής των ελέγχων.

Με την κατάλληλη προετοιμασία, και με την χρήση κατάλληλων εργαλείων, οι έλεγχοι λογισμικού στο νέφος μπορεί να φέρουν πραγματική διαφορά στην ορθότητα, την απόδοση, και το χρηματικό κόστος του σταδίου ελέγχων λογισμικού στον κύκλο ανάπτυξης ενός έργου λογισμικού.

11. Βιβλιογραφία

Cloud και μοντέλα ανάπτυξης

- <https://azure.microsoft.com/en-us/overview/what-is-the-cloud/>
- https://en.wikipedia.org/wiki/Cloud_computing
- Advances in Computers, 2012, by Mohammad Hamdaqa, Ladan Tahvildari, Section 2.2
- <https://www.leadingedgetech.co.uk/it-services/it-consultancy-services/cloud-computing/what-are-the-types-of-cloud-computing/>
- <https://www.javatpoint.com/types-of-cloud>

Cloud Computing

- <https://info.cloudcarib.com/blog/cloud-storage-vs.-cloud-computing-whats-the-difference>
- <https://www.javatpoint.com/cloud-computing-architecture>
- <https://www.leadingedgetech.co.uk/it-services/it-consultancy-services/cloud-computing/what-are-the-types-of-cloud-computing/>
- <https://www.softwaretestinghelp.com/getting-started-with-cloud-testing/>
<https://blog.udemy.com/what-is-cloud-architecture-understanding-the-fundamentals/>
- <https://www.javatpoint.com/infrastructure-as-a-service>
- <https://www.javatpoint.com/platform-as-a-service>
- <https://www.javatpoint.com/software-as-a-service>

Cloud testing

- <https://performancelabus.com/cloud-testing-features-types/>
- <https://www.sciencedirect.com/science/article/pii/S2213020916301999>
- <https://www.ibm.com/topics/software-testing>

Τεχνικές ελέγχου

- <https://searchstorage.techtarget.com/definition/cloud-testing>
- <https://www.guru99.com/what-is-testing-as-a-service-taas.html>
- <https://www.guru99.com/functional-testing.html>
- <https://www.guru99.com/non-functional-testing.html>
- Cloud based Testing Techniques (CTT), Akash Shrivastva, Shubham Gupta, Rinki Tiwar

Η Χρησιμότητα των Ελέγχων στο Νέφος:

- <https://www.lambdatest.com/blog/cloud-testing-tutorial/>
- <https://www.apriorit.com/dev-blog/548-cloud-based-testing>
- https://www.cigniti.com/wp-content/resources/Dataquest-Cigniti_October%202016.pdf

To Remote Testing:

- <https://www.lambdatest.com/blog/10-biggest-remote-testing-challenges/>

Οι Δυσκολίες των Ελέγχων στο Νέφος:

- <https://www.softwaretestinghelp.com/getting-started-with-cloud-testing/>
- <https://techbeacon.com/enterprise-it/why-cloud-isnt-always-best-answer-4-myths-misconceptions>
- <https://techgenix.com/when-to-avoid-cloud-computing/amp/>

Τα στάδια προετοιμασίας για ελέγχους στο νέφος:

- <https://www.apriorit.com/dev-blog/548-cloud-based-testing>

Εργαλεία Ελέγχων στο νέφος:

- <https://www.softwaretestinghelp.com/cloud-testing-tools/>
- <https://www.cigniti.com/blog/top-7-cloud-based-software-testing-tools/>