



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Πρόγραμμα Μεταπτυχιακών Σπουδών
Επιστήμη και Τεχνολογία
της Πληροφορικής και των Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Η επίδραση της χρήσης των περιεκτών και της επακόλουθης ανάγκης μετασχηματισμού ιδιωτικών νεφροϋπολογιστικών υποδομών.

Ιορδάνης Μιχαήλ
A.M. mcse20024

Καλκαβούρας Κωνσταντίνος
A.M. mcse20025

Εισηγητής: Δρ Βασίλειος Μάμαλης, Καθηγητής

Αθήνα, 2023

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Η επίδραση της χρήσης των περιεκτών και της επακόλουθης ανάγκης μετασχηματισμού ιδιωτικών νεφούπολογιστικών υποδομών.

Ιορδάνης Μιχαήλ

A.M. mcse20024

Καλκαβούρας Κωνσταντίνος

A.M. mcse20025

Εισηγητής:

Βασίλειος Μάμαλης, Καθηγητής

**Εξεταστική
Επιτροπή:**

Καρκαζής Παναγιώτης, Αναπληρωτής Καθηγητής

Μάμαλης Βασίλειος, Καθηγητής

Μπόγρης Αντώνιος, Καθηγητής

**Ημερομηνία
εξέτασης:**

27 / 09 / 2023

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Καλκαβούρας Κωνσταντίνος του Δημητρίου, με αριθμό μητρώου mcse20025 φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών "Επιστήμη και Τεχνολογία της Πληροφορικής και των Υπολογιστών" του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι έξι (6) μήνες και έπειτα από αίτηση μου στη Βιβλιοθήκη και έγκριση του επιβλέποντα καθηγητή.

Ο Δηλών

Ο κάτωθι υπογεγραμμένος Ιορδάνης Μιχαήλ του Παύλου, με αριθμό μητρώου mcse20024 φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών "Επιστήμη και Τεχνολογία της Πληροφορικής και των Υπολογιστών" του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Επιθυμώ την απαγόρευση πρόσβασης στο πλήρες κείμενο της εργασίας μου μέχρι έξι (6) μήνες και έπειτα από αίτηση μου στη Βιβλιοθήκη και έγκριση του επιβλέποντα καθηγητή.

Ο Δηλών

ΕΥΧΑΡΙΣΤΙΕΣ

Με το πέρας της παρούσας διπλωματικής εργασίας ολοκληρώνεται ένα πρόγραμμα σπουδών και ταυτόχρονα μια μακροχρόνια συνεργασία δύο ανθρώπων διαφορετικών, αλλά πιστών σε κοινές πεποιθήσεις και ιδέες.

Ευχαριστούμε ο ένας τον άλλο για την καλή διάθεση, την αστείρευτη υπομονή και το πάθος για επιστημοσύνη.

Ευχαριστούμε το Ελληνικό Κέντρο Θαλασσίων Ερευνών για την εμπιστοσύνη και την πρόσβαση που μας παραχώρησε στη διαχείριση των υποδομών του.

Ευχαριστούμε τον κ. Μάμαλη για την στήριξη και την καθοδήγηση του.

Ευχαριστούμε τις οικογένειες μας και κυρίως τις συντρόφους και τα μικρά μέλη τους, που έδειξαν αμέριστη συμπαράσταση και κατανόηση στην χρονοβόρα προσπάθεια ολοκλήρωσης της εργασίας αυτής.

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία εστιάζει στην αξιοποίηση των σύγχρονων τεχνολογιών υπολογιστικής νέφους σε μικρής κλίμακας ιδιωτικές νεφοϋπολογιστικές υποδομές. Η χρήση εικονικών μηχανών και περιεκτών αναδεικνύεται ως ένας αξιόπιστος και αποδοτικός τρόπος φιλοξενίας εφαρμογών και υπηρεσιών, ενώ η διαδικασία μετάπτωσης τους αποτελεί σύνθετη πρόκληση, ειδικά σε ιδιωτικές υποδομές μικρής κλίμακας. Ο κύριος στόχος της εργασίας είναι η διερεύνηση της δυνατότητας μετάπτωσης μικρής κλίμακας ιδιωτικών νεφοϋπολογιστικών υποδομών φιλοξενίας εικονικών μηχανών σε υποδομές συνφιλοξενίας εικονικών μηχανών και περιεκτών, ενσωματώνοντας όλες τις σύγχρονες λειτουργικότητες των πλατφορμών διαχείρισης και ενορχήστρωσης, σε μία ενιαία υποδομή.

Η εργασία μέσω βιβλιογραφικής ανασκόπησης απαντά σε μία σειρά ερευνητικών ερωτημάτων, όπως της καταγραφής των διαθέσιμων μοντέλων υπηρεσιών και ανάπτυξης της υπολογιστικής νέφους, των τρόπων διάθεσης πόρων και των τεχνολογιών εικονικοποίησης. Επιπροσθέτως, διερευνώνται οι παράγοντες και περιορισμοί που προωθούν τον μετασχηματισμό των υποδομών και εφαρμογών και παρουσιάζονται οι επικρατέστερες μεθοδολογίες μετασχηματισμού. Στο ερευνητικό μέρος της εργασίας εξετάζεται η μελέτη περίπτωσης του μετασχηματισμού της ιδιωτικής υποδομής του Ελληνικού Κέντρου Θαλάσσιων Ερευνών. Επισημαίνονται οι λόγοι που οδήγησαν στην ανάγκη μετασχηματισμού και παρουσιάζονται οι συνθήκες και περιορισμοί που πρέπει να ληφθούν υπόψη. Επίσης, αναφέρονται οι απαιτήσεις του Φορέα και η μεθοδολογία που χρησιμοποιήθηκε για την υλοποίηση του μετασχηματισμού. Τέλος, παρουσιάζονται και αξιολογούνται οι διαθέσιμες τεχνολογικές λύσεις.

Συνολικά, η διπλωματική εργασία προσφέρει μια λεπτομερή μελέτη για τη διαχείριση και μετάπτωση μικρής κλίμακας ιδιωτικών νεφοϋπολογιστικών υποδομών, με σκοπό την αξιοποίηση στο έπακρο των δυνατοτήτων των εικονικών μηχανών και περιεκτών σε αυτά τα περιβάλλοντα. Η μελέτη περίπτωσης και οι προτάσεις που παρέχονται μπορούν να αποτελέσουν βάση για περαιτέρω έρευνες και βελτιώσεις, προκειμένου να βελτιστοποιηθούν οι υποδομές και η αποδοτικότητα των φιλοξενούμενων εφαρμογών και υπηρεσιών.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Υπολογιστική Νέφους

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Υπολογιστική Νέφους, Ιδιωτική Νεφρολογιστική Υποδομή, Μετάπτωση στο Νέφος, Ανάπτυξη Μικρής Κλίμακας, Εικονικοποίηση, Εικονικές Μηχανές, Περιέκτες

ABSTRACT

The present thesis focuses on the utilization of modern cloud computing technologies in small-scale private cloud infrastructures. The use of virtual machines and containers is highlighted as a reliable and efficient way of hosting applications and services, while their migration poses a complex challenge, especially in small-scale private infrastructures. The main objective of this thesis is to explore the possibility of migrating small-scale private cloud infrastructures of virtual machines into co-hosted virtual machines and containers infrastructures, incorporating all the modern functionalities of management and orchestration platforms into a unified infrastructure.

This thesis answers a series of research questions through a literature review, such as the recording of available models of cloud service and cloud deployment models, resource allocation methods and virtualization technologies. Additionally, it examines the factors that promote the transformation of infrastructures and applications, and presents the prevailing transformation methodologies. In the research part of the thesis, the case study of transforming the private infrastructure of the Hellenic Centre for Marine Research is examined. The reasons that led to the need for transformation are highlighted, and the conditions and limitations to be considered are presented. Furthermore, the requirements of the Organization and the methodology used for the implementation of the transformation are mentioned. Finally, the available technological solutions are presented and evaluated.

Overall, the thesis offers a detailed study of the management and migration of small-scale private cloud infrastructures, with the aim of fully leveraging the capabilities of virtual machines and containers in these environments. The case study and the proposals provided can serve as a basis for further research and improvements, in order to optimize the infrastructure and the efficiency of hosted applications and services.

SCIENTIFIC FIELD: Cloud Computing

KEYWORDS: Cloud Computing, Private Cloud Infrastructure, Cloud Migration, Small-scale Deployment, Virtualization, Virtual Machines, Containers

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ	10
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ	13
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....	14
ΚΑΤΑΛΟΓΟΣ ΣΤΙΓΜΙΟΤΥΠΩΝ	15
ΕΙΣΑΓΩΓΗ	22
A. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ	25
ΚΕΦΑΛΑΙΟ 1	25
ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ	25
1.1 Μοντέλα Υπηρεσιών Υπολογιστικού Νέφους.....	25
1.1.1 IaaS (Infrastructure as a Service)	26
1.1.2 PaaS (Platform as a Service)	29
1.1.3 SaaS (Software as a Service).....	31
1.1.4 XaaS (Anything as a Service).....	33
1.2 Μοντέλα Ανάπτυξης Υπολογιστικού Νέφους.....	33
1.2.1 Ιδιωτικό Νέφος (Private Cloud).....	34
1.2.2 Δημόσιο Νέφος (Public Cloud)	36
1.2.3 Νέφος Κοινότητας (Community Cloud).....	37
1.2.4 Υβριδικό Νέφος (Hybrid Cloud)	38
ΚΕΦΑΛΑΙΟ 2	41
ΤΡΟΠΟΙ ΔΙΑΘΕΣΗΣ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΠΟΡΩΝ ΣΤΟ ΝΕΦΟΣ	41
2.1 Εξυπηρετητές Γυμνού Μετάλλου (Bare-Metal Servers)	42
2.2 Εικονικοποίηση.....	43
2.2.1 Είδη εικονικοποίησης.....	44
2.3 Εικονικές Μηχανές.....	44
2.3.1 Υπερεπτόπτες Εικονικών Μηχανών.....	46
2.3.1.1 Υπερεπτόπτες Τύπου 1 (Type 1 Hypervisors).....	47
2.3.1.2 Υπερεπτόπτες Τύπου 2 (Type 2 Hypervisors).....	48
2.4 Περιέκτες	49
2.4.1 Μηχανές και Περιβάλλοντα Εκτέλεσης Περιεκτών	51

2.4.1.1	LXC (LinuX Containers).....	53
2.4.1.2	Docker	57
2.4.1.3	Containerd	61
2.4.1.4	CRI-O	62
2.4.1.5	runC.....	62
2.4.2	Ενορχηστρωτές Περιεκτών	63
2.4.3	Πλατφόρμες Περιεκτών.....	69
ΚΕΦΑΛΑΙΟ 3.....		71
ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ ΕΦΑΡΜΟΓΩΝ ΚΑΙ ΥΠΟΔΟΜΩΝ ΣΤΑ ΣΥΓΧΡΟΝΑ		
ΠΡΟΤΥΠΑ ΠΕΡΙΕΚΤΩΝ.....		71
3.1	Μετασχηματισμός υποδομής φιλοξενίας εικονικών μηχανών σε υποδομή συνφιλοξενίας περιεκτών	71
3.1.1	Παράλληλη υποδομή (Side to side).....	72
3.1.2	Ενιαία υποδομή (All in One)	73
3.1.3	Υβριδική υποδομή (Hybrid - CaaS over IaaS).....	75
3.2	Μετασχηματισμός εφαρμογών και ενσωμάτωση τους σε περιέκτες.....	78
3.2.1	Παράγοντες και περιορισμοί μετασχηματισμού και ενσωμάτωσης εφαρμογών σε περιέκτες.....	78
3.2.2	Μεθοδολογίες μετασχηματισμού εικονικών μηχανών και εφαρμογών σε περιέκτες (on-cloud containerization).....	80
3.2.2.1	Rehost (Lift-and-Shift)	81
3.2.2.2	Replatform ("Lift and reshape", "Lift, Tinker and Shift").....	82
3.2.2.3	Refactor / Rearchitect.....	82
B. ΕΡΕΥΝΗΤΙΚΟ ΜΕΡΟΣ		84
ΚΕΦΑΛΑΙΟ 4.....		84
ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ (CASE STUDY)		84
4.1	Η περίπτωση του Ελληνικού Κέντρου Θαλάσσιων Ερευνών (ΕΛ.ΚΕ.Θ.Ε.) 85	
4.2	Περίληψη – Περιγραφή προβλήματος	87
4.3	Μελέτη υπάρχουσας κατάστασης και Εκτίμηση λύσεων	88
4.3.1	Ανάλυση απαιτήσεων	90
4.3.1.1	Ανάλυση Λειτουργικών Απαιτήσεων ΕΛΚΕΘΕ.....	90
4.3.1.2	Ανάλυση Τεχνολογικών Περιορισμών ΕΛΚΕΘΕ	92
4.3.1.3	Ανάλυση των προτιμήσεων και δυνατοτήτων των χρηστών ΕΛΚΕΘΕ.....	94
4.3.2	Βιβλιογραφική Ανασκόπηση	96

4.3.3	Τεχνική Αξιολόγηση	97
4.3.3.1	Κατανόηση της υφιστάμενης τεχνολογίας	97
4.3.3.2	Εξέταση διαθέσιμων λύσεων	98
4.3.3.3	Εκτίμηση απαιτήσεων της κάθε λύσης	105
4.3.3.4	Αξιολόγηση της Συμβατότητας	109
4.3.4	Οικονομική Ανάλυση.....	115
4.3.5	Επιλογή συμβατών λύσεων	117
4.4	Υλοποίηση και παρουσίαση λύσεων	118
4.4.1	Προετοιμασία υποδομής και Παραδοχές	119
4.4.2	Rancher, με Harvester με διάφορες διανομές Kubernetes	120
4.4.2.1	Εγκατάσταση υποδομής.....	121
4.4.2.2	Προσβασιμότητα στο UI	125
4.4.2.3	Δημιουργία χρηστών και διαχείριση πόρων	126
4.4.2.4	Εγκατάσταση δοκιμαστικής εφαρμογής.....	130
4.4.2.5	Επέκταση υποδομής	136
4.4.2.6	Αναβάθμιση υποδομής.....	139
4.4.2.7	Έλεγχος για υψηλή διαθεσιμότητα	142
4.4.2.8	Δοκιμή ολικής απενεργοποίησης και επαναφοράς.....	146
4.4.3	OKD αξιοποιώντας την δυνατότητα του για ενσωμάτωση στο oVirt.....	148
4.4.3.1	Εγκατάσταση υποδομής.....	148
4.4.3.2	Προσβασιμότητα στο UI	151
4.4.3.3	Δημιουργία χρηστών και διαχείριση πόρων	153
4.4.3.4	Εγκατάσταση δοκιμαστικής εφαρμογής.....	155
4.4.3.5	Επέκταση υποδομής	169
4.4.3.6	Αναβάθμιση υποδομής.....	172
4.4.3.7	Έλεγχος για υψηλή διαθεσιμότητα	176
4.4.3.8	Δοκιμή ολικής απενεργοποίησης και επαναφοράς.....	183
4.5	Συμπεράσματα και Αξιολόγηση λύσεων.....	184
ΚΕΦΑΛΑΙΟ 5		188
ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ.....		188
5.1	Ανασκόπηση Στόχων και Ανάλυση Ευρημάτων	188
5.2	Περιορισμοί της Έρευνας	189
5.3	Ερευνητικό Κενό και Προτάσεις για Μελλοντική Έρευνα.....	191
5.4	Συμπεράσματα	193

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1. Τρόποι διάθεσης πόρων	42
Σχήμα 2. Υπερεπώπτες εικονικών μηχανών	47
Σχήμα 3. Απεικόνιση αρχιτεκτονικής συστοιχίας ενορχήστρωσης περιεκτών	73
Σχήμα 4. Απεικόνιση αρχιτεκτονικής συστοιχίας διαχειριστή εικονικών μηχανών	73
Σχήμα 5. Απεικόνιση αρχιτεκτονικής συστοιχίας ενιαίας πλατφόρμας διαχείρισης εικονικών μηχανών και περιεκτών	75
Σχήμα 6. Απεικόνιση αρχιτεκτονικής συστοιχίας ενορχηστρωτή περιεκτών σε συστοιχία διαχείρισης εικονικών μηχανών.....	76
Σχήμα 7. Μεθοδολογία "Lift & Shift"	81
Σχήμα 8. Μεθοδολογία "Replatform"	82
Σχήμα 9. Μεθοδολογία "Refactor"	83
Σχήμα 10. Σχηματική απεικόνιση διαδικασίας ολικής απενεργοποίησης και επαναφοράς της υποδομής Harvester και Rancher	147
Σχήμα 11. Σχηματική απεικόνιση διαδικασίας ελέγχου υψηλής διαθεσιμότητας OKD.....	183

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1. Μοντέλα υπηρεσιών υπολογιστικής νέφους [6]	26
Εικόνα 2. Μοντέλα ανάπτυξης υπολογιστικής νέφους [14].....	34
Εικόνα 3. Πίνακας συμβατότητας εκδόσεων Rancher, Harvester και K8S.....	140
Εικόνα 4. Αρχιτεκτονική συστοιχίας OKD σε συστοιχία oVirt	150
Εικόνα 5. Πίνακας συμβατότητας εκδόσεων OKD και RHV	173
Εικόνα 6. Πίνακας κόμβων etcd με βάση τον αλγόριθμο ομοφωνίας Raft	177

ΚΑΤΑΛΟΓΟΣ ΣΤΙΓΜΙΟΤΥΠΩΝ

Στιγμιότυπο 1. Εικονικές μηχανές υποδομής Harvester	122
Στιγμιότυπο 2. Οθόνη επιτυχούς φόρτωσης λογισμικού Harvester	122
Στιγμιότυπο 3. Εικονική μηχανή για την εγκατάσταση του Rancher	123
Στιγμιότυπο 4. Ενσωμάτωση Harvester στο Rancher (1).....	124
Στιγμιότυπο 5. Ενσωμάτωση Harvester στο Rancher (2).....	125
Στιγμιότυπο 6. Ολοκλήρωση διαδικασίας ενσωμάτωσης Harvester στο Rancher	125
Στιγμιότυπο 7. Harvester: Γραφική διεπαφή.....	126
Στιγμιότυπο 8. Rancher: Γραφική διεπαφή.....	126
Στιγμιότυπο 9. Rancher: Διεπαφή χρηστών	127
Στιγμιότυπο 10. Rancher: Διασύνδεση Kubernetes συστοιχίας με χρήστη	128
Στιγμιότυπο 11. Rancher: Λίστα Kubernetes συστοιχιών	128
Στιγμιότυπο 12. Δημιουργία νέας Kubernetes συστοιχίας στο Harvester μέσω της διεπαφής του Rancher (1)	129
Στιγμιότυπο 13. Δημιουργία νέας Kubernetes συστοιχίας στο Harvester μέσω της διεπαφής του Rancher (2)	129
Στιγμιότυπο 14. Δημιουργία νέας Kubernetes συστοιχίας στο Harvester μέσω της διεπαφής του Rancher (3)	130
Στιγμιότυπο 15. Δημιουργία νέας Kubernetes συστοιχίας στο Harvester μέσω της διεπαφής του Rancher (4)	130
Στιγμιότυπο 16. Rancher: Επιλογή συστοιχίας για την εγκατάσταση δοκιμαστικών εφαρμογών	131
Στιγμιότυπο 17. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (1).....	131
Στιγμιότυπο 18. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (2).....	132
Στιγμιότυπο 19. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (3).....	132
Στιγμιότυπο 20. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (4).....	133
Στιγμιότυπο 21. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (5).....	133

Στιγμιότυπο 22. Rancher: Έλεγχος λειτουργίας βάσης δεδομένων MariaDB.....	134
Στιγμιότυπο 23. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (1).....	135
Στιγμιότυπο 24. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (2).....	135
Στιγμιότυπο 25. Έλεγχος λειτουργίας εξυπηρετητή δικτύου (webserver) nginx .	136
Στιγμιότυπο 26. Επέκταση υποδομής Harvester.....	137
Στιγμιότυπο 27. Rancher: Επέκταση Kubernetes συστοιχίας με προσθήκη επιπλέον κόμβων (1)	137
Στιγμιότυπο 28. Rancher: Επέκταση Kubernetes συστοιχίας με προσθήκη επιπλέον κόμβων (2)	138
Στιγμιότυπο 29. Rancher: Επέκταση Kubernetes συστοιχίας με προσθήκη επιπλέον κόμβων (3)	138
Στιγμιότυπο 30. Rancher: Επέκταση Kubernetes συστοιχίας με προσθήκη επιπλέον κόμβων (4)	139
Στιγμιότυπο 31. Αναβάθμιση Harvester μέσω γραφικής διεπαφής χρήστη (1) ..	141
Στιγμιότυπο 32. Αναβάθμιση Harvester μέσω γραφικής διεπαφής χρήστη (2) ..	141
Στιγμιότυπο 33. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (1)	142
Στιγμιότυπο 34. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (2)	142
Στιγμιότυπο 35. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (3)	143
Στιγμιότυπο 36. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (4)	143
Στιγμιότυπο 37. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (5)	143
Στιγμιότυπο 38. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (6)	144
Στιγμιότυπο 39. Δοκιμή υψηλής διαθεσιμότητας συστοιχιών Kubernetes της υποδομής Rancher (1)	145
Στιγμιότυπο 40. Δοκιμή υψηλής διαθεσιμότητας συστοιχιών Kubernetes της υποδομής Rancher (2)	145
Στιγμιότυπο 41. Εικονικές μηχανές συστοιχίας OKD στο oVirt.....	151
Στιγμιότυπο 42. Γραφικό περιβάλλον OKD - κονσόλα διαχειριστή.....	152
Στιγμιότυπο 43. Γραφικό περιβάλλον OKD - κονσόλα προγραμματιστή.....	152
Εν συνέχεια, η ομάδα περιηγήθηκε στις διάφορες διεπαφές που διαθέτει το γραφικό περιβάλλον, το οποίο διαχωρίζεται σε δύο κύριες κατηγορίες διεπαφών, του Διαχειριστή (Administrator Console) της συστοιχίας και του Προγραμματιστή (Developer Console) (βλ. Στιγμιότυπο 42-43).	152

Στιγμιότυπο 44. OKD: Διεπαφή προβολής χρηστών	154
Στιγμιότυπο 45. OKD: Διεπαφή προσθήκης μηχανισμού αυθεντικοποίησης	154
Στιγμιότυπο 46. OKD: Ανάθεση πόρων σε έργο	155
Στιγμιότυπο 47. OKD: Χρήση πόρων	155
Η διαδικασία της εγκατάστασης της βάσης δεδομένων υλοποιήθηκε με χρήση των διαθέσιμων διεπαφών που διαθέτει η διεπαφή Προγραμματιστή (Developer Console), και αξιοποιώντας τα προεγκατεστημένα αποθετήρια εφαρμογών. Για την απομόνωση και διαχωρισμό των χρησιμοποιούμενων πόρων, δημιουργήθηκε ξεχωριστό project (namespace) εντός του οποίου ομαδοποιούνται όλοι οι πόροι που απαιτούνται από την εφαρμογή. Τα βήματα που απαιτούνται για την εγκατάσταση της δοκιμαστικής εφαρμογής MariaDB προκύπτουν εύκολα και με σαφήνεια από το γραφικό περιβάλλον διευκολύνοντας και καθοδηγώντας τον χρήστη (βλ. Στιγμιότυπο 48).....	156
Στιγμιότυπο 48. Εγκατάσταση βάσης δεδομένων MariaDB μέσω αποθετηρίου Helm (1).....	157
Στιγμιότυπο 49. Εγκατάσταση βάσης δεδομένων MariaDB μέσω αποθετηρίου Helm (2).....	157
Στιγμιότυπο 50. Παρακολούθηση γεγονότων εγκατάστασης βάσης δεδομένων MariaDB.....	158
Στιγμιότυπο 51. Γραφική απεικόνιση βάσης δεδομένων MariaDB στην OKD διεπαφή	158
Στιγμιότυπο 52. Προβολή των πόρων της εφαρμογής MariaDB στη διεπαφή του OKD	159
Στιγμιότυπο 53. Έλεγχος εγκατάστασης βάσης δεδομένων MariaDB.....	159
Στιγμιότυπο 54. Έλεγχος ορθής λειτουργίας βάσης δεδομένων MariaDB	160
Στιγμιότυπο 55. Εγκατάσταση της βοηθητικής εφαρμογής phpMyAdmin (1).....	160
Στιγμιότυπο 56. Εγκατάσταση της βοηθητικής εφαρμογής phpMyAdmin (2).....	161
Στιγμιότυπο 57. Έλεγχος ορθής λειτουργίας phpMyAdmin	161
Στιγμιότυπο 58. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (1).....	162
Στιγμιότυπο 59. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (2).....	162
Στιγμιότυπο 60. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (3).....	163

Στιγμιότυπο 61. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (4).....	164
Στιγμιότυπο 62. Προβολή των πόρων της εφαρμογής εξυπηρετητή δικτύου (webserver) nginx	165
Στιγμιότυπο 63. Έλεγχος ορθής λειτουργίας της εφαρμογής εξυπηρετητή δικτύου (webserver) nginx (1).....	165
Στιγμιότυπο 64. Έλεγχος ορθής λειτουργίας της εφαρμογής εξυπηρετητή δικτύου (webserver) nginx (2).....	166
Στιγμιότυπο 65. Εγκατάσταση WordPress εφαρμογής μέσω αρχείου yaml (1)..	168
Στιγμιότυπο 66. Εγκατάσταση WordPress εφαρμογής μέσω αρχείου yaml (2)..	168
Στιγμιότυπο 67. Εγκατάσταση WordPress εφαρμογής μέσω αρχείου yaml (3)..	169
Στιγμιότυπο 68. Οριζόντια επέκταση υποδομής (1)	170
Στιγμιότυπο 69. Οριζόντια επέκταση υποδομής (2)	171
Στιγμιότυπο 70. Οριζόντια επέκταση υποδομής (3)	171
Στιγμιότυπο 71. Κάθετη επέκταση υποδομής (1)	171
Στιγμιότυπο 72. Κάθετη επέκταση υποδομής (2)	172
Στιγμιότυπο 73. Προβολή διαθέσιμης αναβάθμισης OKD (1).....	174
Στιγμιότυπο 74. Προβολή διαθέσιμης αναβάθμισης OKD (2).....	175
Στιγμιότυπο 75. Προβολή διαθέσιμης αναβάθμισης OKD (3).....	175
Στιγμιότυπο 76. Προβολή διαθέσιμης αναβάθμισης OKD μέσω γραμμής εντολών	176
Στιγμιότυπο 77. Έλεγχος υψηλής διαθεσιμότητας OKD (1)	177
Στιγμιότυπο 78. Έλεγχος υψηλής διαθεσιμότητας OKD (2)	178
Στιγμιότυπο 79. Έλεγχος υψηλής διαθεσιμότητας OKD (3)	178
Στιγμιότυπο 80. Έλεγχος υψηλής διαθεσιμότητας OKD (4)	178
Στιγμιότυπο 81. Έλεγχος κόμβου εφαρμογής nginx.....	180
Στιγμιότυπο 82. Έλεγχος υψηλής διαθεσιμότητας OKD (5)	180
Στιγμιότυπο 83. Έλεγχος επαναφοράς εφαρμογής σε εναπομείναντα κόμβο (1)	181
Στιγμιότυπο 84. Έλεγχος υψηλής διαθεσιμότητας OKD (6)	181
Στιγμιότυπο 85. Έλεγχος επαναφοράς εφαρμογής σε εναπομείναντα κόμβο (2)	182
Στιγμιότυπο 86. Έλεγχος υψηλής διαθεσιμότητας OKD (7)	182
Στιγμιότυπο 87. Δοκιμή ολικής απενεργοποίησης και επαναφοράς (1)	184

Στιγμιότυπο 88. Δοκιμή ολικής απενεργοποίησης και επαναφοράς (2) 184

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

AKS Azure Kubernetes Services

API Application Programming Interface

AppArmor Application Armor

AUFS Advanced Multi-Layered Unification Filesystem

Btrfs B-tree filesystem

CaaS Container as a Service

CAPEX Capital Expenditure

CI/CD Continuous Integration / Continuous Development

CLI Command Line Interface

CNCF Cloud Native Computing Foundation

CoW Copy on Write

CPU Central Processing Unit

CRD Custom Resource Definition

CRI Container Runtime Interface

CSP Cloud Service Provider

CSR Certificate Signing Request

DaaS Desktop as a Service

DBaaS DataBase as a Service

DR Disaster Recovery

EC2 Amazon Elastic Compute Cloud

GCE Google Compute Engine

HCI Hyper-Converged Infrastructure

HCL HashiCorp Configuration Language

HPC High Performance Computing

HTTP/HTTPS HyperText Transfer Protocol / HyperText Transfer Protocol Secure

IDS Intrusion Detection Systems

IP Internet Protocol

K3s Lightweight Kubernetes

KVM Kernel-based Virtual Machine

LXC LinuX Containers

MicroK8s Micro Kubernetes

OC OpenShift command-line interface
OCI Open Container Initiative
OS Operating System
PCI Peripheral Component Interface
PVC Persistent Volume Claim
QEMU Quick Emulation
CI/CD Continuous Integration / Continuous Delivery
RAID Redundant Array of Inexpensive Disks
RAM Random Access Memory
RKE Rancher Kubernetes Engine
RKE Rancher Kubernetes Engine
SECaaS Security as a Service
SELinux Security-Enhanced Linux
SLA Service Level Agreement
SOA Service Oriented Architecture
SR-IOV Single Root I/O Virtualization
SSO Single Sign-On
STaaS Storage as a Service
SWOT Strengths, Weaknesses, Opportunities, Threats
VC Volume Claim
VE Virtual Environment
VLAN Virtual Local Area Network
VM Virtual Machine
VMM Virtual Machine Manager
XaaS Anything as a Service
KME Κεντρική Μονάδα Επεξεργασίας

ΕΙΣΑΓΩΓΗ

Η τεχνολογία της εικονικοποίησης έχει αποτελέσει μια επαναστατική προσέγγιση στον κόσμο της πληροφορικής και έχει συνδεθεί άρρηκτα με την υπολογιστική νέφος. Καθώς οι επιχειρήσεις και οι οργανισμοί επιδιώκουν να βελτιώσουν την αποτελεσματικότητα και την ευελιξία των υποδομών τους, η χρήση εικονικών μηχανών και περιεκτών κρίνεται ως ένας αξιόπιστος και αποδοτικός τρόπος φιλοξενίας εφαρμογών και υπηρεσιών. Παράλληλα, η επικρατούσα τάση ανάπτυξης εφαρμογών, στα πρότυπα της αρχιτεκτονικής νέφος και των μικροϋπηρεσιών, έχουν γιγαντώσει την χρήση των περιεκτών, αφού αποτελούν την συνηθέστερη μορφή διανομής τους. Επιπρόσθετα, οι τεχνικές της συνεχούς ενσωμάτωσης και ανάπτυξης (CI/CD) ενδυναμώνουν την αξιοποίηση των περιεκτών και προσελκύουν τη χρήση υπηρεσιών πλατφόρμας (PaaS) για την διαχείριση τους. Οι παραπάνω συνθήκες δημιουργούν την ανάγκη ολοκληρωμένης υποστήριξης εικονικών μηχανών και περιεκτών στις ιδιωτικές υποδομές επιχειρήσεων και οργανισμών και κατά συνέπεια την ανάγκη μετασχηματισμού τους. Ωστόσο, η διαδικασία μετάπτωσης υποδομών που υποστηρίζουν εικονικές μηχανές σε υποδομές που υποστηρίζουν παράλληλα εικονικές μηχανές και περιέκτες είναι σύνθετη και μπορεί να αποτελέσει μεγάλη πρόκληση, ιδιαίτερα στην περίπτωση που αφορά ιδιωτικές υποδομές μικρής κλίμακας.

Ο κύριος στόχος της παρούσας διπλωματικής εργασίας είναι η διερεύνηση της δυνατότητας μετάπτωσης μικρής κλίμακας ιδιωτικών νεφοϋπολογιστικών υποδομών φιλοξενίας εικονικών μηχανών σε υποδομές συνφιλοξενίας εικονικών μηχανών και περιεκτών, ενσωματώνοντας όλες τις σύγχρονες λειτουργικότητες των πλατφορμών διαχείρισης και ενορχήστρωσης, σε μία ενιαία υποδομή. Πιο συγκεκριμένα, θα μελετηθούν οι τεχνολογίες εικονικοποίησης, οι μεθοδολογίες μετασχηματισμού υποδομών και εφαρμογών, καθώς και η πρακτική εφαρμογή της διαδικασίας μετασχηματισμού ιδιωτικών υποδομών.

Για την κατανόηση και την αξιολόγηση της διαδικασίας μετάπτωσης μπορεί να προσδιοριστεί μια σειρά ερευνητικών ερωτημάτων. Το αρχικό ερώτημα αφορά την καταγραφή των διαθέσιμων μοντέλων υπηρεσιών και των μοντέλων ανάπτυξης της υπολογιστικής νέφος. Επίσης αναζητούνται οι τρόποι διάθεσης πόρων και εξετάζονται οι τεχνολογίες εικονικοποίησης. Ακολούθως διερευνώνται οι

παράγοντες που περιορίζουν τον μετασχηματισμό των υποδομών και των εφαρμογών, ενώ καταγράφονται οι επικρατέστερες μεθοδολογίες μετασχηματισμού. Στη συνέχεια εξετάζεται η δυνατότητα πρακτικής εφαρμογής των μεθοδολογιών μετασχηματισμού, συνδυαστικά με την εξεύρεση των διαθέσιμων τεχνολογιών και λογισμικών, με σκοπό την ολοκληρωμένη μετάπτωση μικρής κλίμακας υποδομών ενός οργανισμού ή μιας επιχείρησης. Τέλος, παρουσιάζονται τα οφέλη, τα προβλήματα και οι περιορισμοί της διαδικασίας μετασχηματισμού ιδιωτικών υποδομών.

Στα πλαίσια της μελέτης των στόχων και των ερωτημάτων που προέκυψαν, η παρούσα εργασία χωρίζεται σε τρία μέρη, το θεωρητικό μέρος, το ερευνητικό μέρος και τα συμπεράσματα / συζήτηση.

Αναλυτικότερα, το θεωρητικό μέρος αποτελείται από τρία κεφάλαια. Στο πρώτο, που φέρει τον τίτλο «Νεφοϋπολογιστικές Υποδομές – Υπολογιστικό Νέφος» γίνεται μια εισαγωγική αναφορά στην Υπολογιστική Νέφους και παρουσιάζονται τα μοντέλα υπηρεσιών και τα μοντέλα ανάπτυξης της. Στο δεύτερο κεφάλαιο αντίστοιχα, με τον τίτλο «Τρόποι Διάθεσης Πόρων», αποσαφηνίζονται οι τεχνικές διάθεσης υπολογιστικών πόρων, γίνεται αναφορά στις τεχνολογίες εικονικοποίησης, ενώ αναλύονται εκτενώς οι τεχνολογίες των περιεκτών και των πλατφορμών ενορχήστρωσης τους. Τέλος, στο τρίτο κεφάλαιο του θεωρητικού μέρους, με τίτλο «Μετασχηματισμός Εφαρμογών και Υποδομών στα Σύγχρονα Πρότυπα Περιεκτών», διερευνώνται οι παράγοντες που οδηγούν ή καθορίζουν τους μετασχηματισμούς αυτούς, ενώ παράλληλα καταγράφονται οι κύριες μεθοδολογίες μετασχηματισμού υποδομών και μετασχηματισμού εικονικών μηχανών σε περιέκτες.

Ακολουθεί το ερευνητικό μέρος της παρούσας εργασίας, όπου εξετάζεται η μελέτη περίπτωσης μετασχηματισμού της ιδιωτικής υποδομής του Ελληνικού Κέντρου Θαλάσσιων Ερευνών. Αρχικά, αναλύονται οι λόγοι που ανέδειξαν την ανάγκη μετασχηματισμού και παράλληλα περιγράφονται οι συνθήκες στις οποίες πρέπει να εκπονηθεί ο μετασχηματισμός, περικλείοντας σχετικούς περιορισμούς και παραδοχές. Στη συνέχεια καταγράφονται οι απαιτήσεις του Φορέα και παρουσιάζεται η μεθοδολογική προσέγγιση που ακολούθησε η ομάδα διαχείρισης συστημάτων και δικτύων, για την υλοποίηση του μετασχηματισμού. Ακολούθως εξετάζονται αναλυτικά οι διαθέσιμες τεχνολογικές λύσεις και παρουσιάζονται

λεπτομερώς οι δοκιμές που εκπονήθηκαν στις προαχθείσες λύσεις. Καταληκτικά, παρατίθενται η αξιολόγηση των τεχνολογικών λύσεων και τα συμπεράσματα που παρήχθησαν από την ομάδα διαχείρισης συστημάτων και δικτύων, μετά την υλοποίηση των λύσεων αυτών.

Τέλος, στο τρίτο μέρος της μελέτης γίνεται ανασκόπηση των στόχων και ανάλυση των ευρημάτων, καταγράφονται τυχόν περιορισμοί, αλλά και προτάσεις για μελλοντικές μελέτες και βελτιώσεις, προκειμένου να αξιοποιηθούν στο έπακρο οι δυνατότητες των εικονικών μηχανών και των περιεκτών σε περιβάλλοντα ιδιωτικών νεφοϋπολογιστικών υποδομών.

Συνοψίζοντας, η παρούσα διπλωματική εργασία στοχεύει στην παροχή ενός συνόλου πληροφοριών για τη διαχείριση και την μετάπτωση μικρής κλίμακας ιδιωτικών νεφοϋπολογιστικών υποδομών. Η διερεύνηση της μελέτης περίπτωσης, σε συνδυασμό με τα συμπεράσματα που θα εξαχθούν, μπορούν να διαμορφώσουν την βάση για περαιτέρω έρευνες και προσεγγίσεις, που θα στοχεύουν στη βελτιστοποίηση των υποδομών και την ενίσχυση της αποδοτικότητας και αξιοπιστίας των εφαρμογών και υπηρεσιών που φιλοξενούνται σε αυτές.

A. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

ΚΕΦΑΛΑΙΟ 1

Υπολογιστικό Νέφος

Σύμφωνα με τον ορισμό του NIST [1], το Υπολογιστικό Νέφος είναι ένα μοντέλο το οποίο επιτρέπει την καθολική, εύκολη και κατά απαίτηση πρόσβαση σε μια κοινή δεξαμενή διαμορφώσιμων υπολογιστικών πόρων (π.χ. δίκτυα, εξυπηρετητές, χώρους αποθήκευσης, εφαρμογές και υπηρεσίες). Οι διαμορφώσιμοι υπολογιστικοί πόροι πρέπει να μπορούν να προσφέρονται και να διατίθενται γρήγορα, με ελάχιστη διαχειριστική προσπάθεια ή διάδραση του Παρόχου.

Για την πραγμάτωση των παραπάνω απαιτήσεων και γενικότερα της Υπολογιστικής Νέφους καθοριστικό ρόλο έπαιξαν τεχνολογίες [2] όπως η εικονικοποίηση (virtualization), η υπηρεσιο-κεντρική αρχιτεκτονική (SOA – Service Oriented Architecture), η ευρυζωνικότητα των δικτύων, ο Ιστός 2.0 και άλλες.

Το μοντέλο της Υπολογιστικής Νέφους αποτελείται από πέντε στοιχειώδη χαρακτηριστικά, τρία μοντέλα υπηρεσιών και τέσσερα μοντέλα ανάπτυξης.

Επιγραμματικά τα πέντε στοιχειώδη χαρακτηριστικά είναι

1. Κατά απαίτηση αυτό εξυπηρετήση
2. Ευρεία δικτυακή πρόσβαση
3. Δεξαμενοποίηση πόρων (Resource Pooling)
4. Γρήγορη ελαστικότητα
5. Μετρήσιμες υπηρεσίες

Στις επόμενες ενότητες παρουσιάζονται τα μοντέλα υπηρεσιών και τα μοντέλα ανάπτυξης.

1.1 Μοντέλα Υπηρεσιών Υπολογιστικού Νέφους

Στο NIST μοντέλο, ορίζονται τα ακόλουθα τρία μοντέλα υπηρεσιών της υπολογιστικής νέφους:

1. SaaS (Software as a Service - Λογισμικό ως Υπηρεσία)

2. PaaS (Platform as a Service – Πλατφόρμα ως Υπηρεσία)

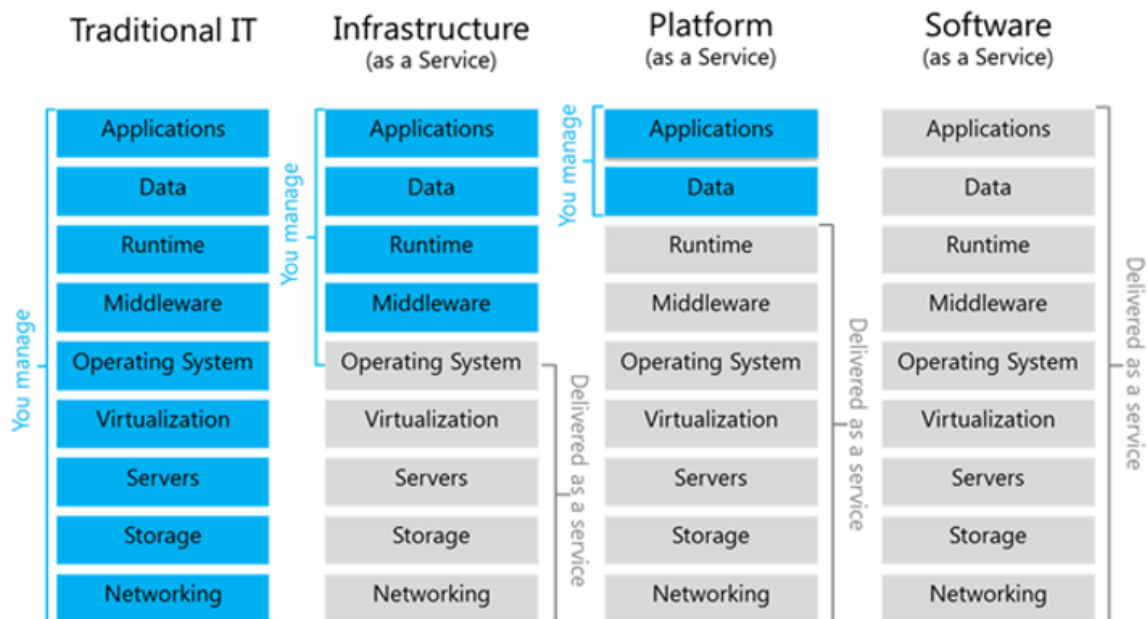
3. IaaS (Infrastructure as a Service – Υποδομή ως Υπηρεσία)

Επιπρόσθετα γίνεται αναφορά και στο μοντέλο XaaS που αποτελεί έναν γενικευμένο όρο σχετικά με τα πιθανά διαθέσιμα μοντέλα υπηρεσιών.

- XaaS [3] [4] [5] (Anything as a Service – Οτιδήποτε ως Υπηρεσία)

Το IaaS μοντέλο υπηρεσίας είναι η βάση για την προσφορά των υπόλοιπων μοντέλων υπηρεσίας. Προσφέρει νεφοϋπολογιστική υποδομή στην οποία μπορούν να λειτουργήσουν το PaaS και το SaaS. Στο PaaS επιπρόσθετα της υποκείμενης υποδομής οι Πάροχοι Υπηρεσιών Νέφους (Cloud Service Provider - CSP) διαχειρίζονται και προσφέρουν λειτουργικά συστήματα, ενδιάμεσο λογισμικό και εργαλεία ανάπτυξης λογισμικού. Το SaaS βρίσκεται στην κορυφή της στοίβας των μοντέλων υπηρεσιών και προσφέρει ολοκληρωμένες εφαρμογές λογισμικού για την ανάπτυξη και την συντήρηση των οποίων είναι εξ' ολοκλήρου υπεύθυνος ο πάροχος υπηρεσιών νέφους.

Ακολουθεί μία πιο αναλυτική περιγραφή των τριών αυτών μοντέλων υπηρεσίας.



Εικόνα 1. Μοντέλα υπηρεσιών υπολογιστικής νέφους [6]

1.1.1 IaaS (Infrastructure as a Service)

Το IaaS μοντέλο υπηρεσίας προσφέρει στους χρήστες τη δυνατότητα χρήσης, κατά απαίτηση, της νεφοϋπολογιστικής υποδομής του παρόχου υπηρεσιών

νέφους. Το IaaS προσφέρει τις περισσότερες επιλογές διαχείρισης σε σύγκριση με τα άλλα δύο Μοντέλα Υπηρεσιών και αντικαθιστά μια τοπικά εγκατεστημένη εταιρική υποδομή με διαφανείς υποδομές σε απομακρυσμένα νέφη με υποστήριξη καθολικής πρόσβασης. Οι ελάχιστες απαιτήσεις για τη δημιουργία μιας IaaS υποδομής περιλαμβάνουν έναν Υπερεπόπτη (Hypervisor - Virtual Machine Manager - VMM) και μία δικτυακή τοπολογία, που μπορεί να είναι είτε ιδιωτική, είτε δημόσια.

Το IaaS μπορεί να θεωρηθεί ως η βάση και θεμέλιος λίθος των μοντέλων υπηρεσιών της Υπολογιστικής Νέφους. Παρότι είναι η βάση και λειτουργεί σαν υπόστρωμα όλων των Μοντέλων Υπηρεσιών, δεν είναι η υπηρεσία που προσφέρθηκε πρώτη στους χρήστες, ούτε αυτή που έχει τη μεγαλύτερη απήχηση. Πολλές φορές το IaaS αναφέρεται και ως HaaS δηλαδή Υλικό ως Υπηρεσία (Hardware as a Service). [7] [8]

Στο IaaS, οι πάροχοι υπηρεσιών νέφους παρέχουν πόρους με τη μορφή Εικονικών Μηχανών (Virtual Machines - VMs), οι οποίες ελέγχονται από κάποιον υπερεπόπτη. Οι πάροχοι παρακολουθούν (monitor) τους υπερεπόπτες και οι πελάτες νοικιάζουν πρόσβαση στο φιλοξενούμενο Λειτουργικό Σύστημα (Operating System - OS) το οποίο εκτελείται σε κάποια εικονική μηχανή.

Ένα IaaS περιβάλλον αποτελείται από έξι επίπεδα [5],

6. Επίπεδο φιλοξενούμενων εφαρμογών και δεδομένων
5. Επίπεδο φιλοξενούμενου Λειτουργικού Συστήματος
4. Επίπεδο Εικονικοποίησης
3. Επίπεδο Host
2. Επίπεδο Φυσικού Υλικού (HW)
1. Επίπεδο Δικτύου

Με το IaaS, ο πελάτης δεν χρειάζεται να ελέγχει ή να διαχειρίζεται όλα τα επίπεδα της υποδομής του, ασχολείται μόνο με τη διαχείριση του φιλοξενούμενου Λειτουργικού Συστήματος (Επίπεδο 5) και των φιλοξενούμενων εφαρμογών που ο ίδιος έχει αναπτύξει (Επίπεδο 6).

Το IaaS μοντέλο υπηρεσίας εξαλείφει την ανάγκη των πελατών για φυσική τοπική ύπαρξη κάποιου Κέντρου Δεδομένων (Data Center) και κατ' επέκταση την ανάγκη συντήρησης και διαχείρισης του. Λειτουργώντας στα πλαίσια πολυχρηστικής (multitenant) αρχιτεκτονικής δίνει την ευκαιρία στους πελάτες να εκσυγχρονίζουν (upgrade) και να επεκτείνουν (scale) τις υπολογιστικές και τις δικτυακές δυνατότητες τους, χωρίς να χρειάζεται να επενδύσουν σε νεφοϋπολογιστικούς πόρους, ελαχιστοποιώντας, μεταξύ άλλων, το Κεφάλαιο Αρχικής Επένδυσης (CAPEX - Capital Expenditure). [9] [10]

Ο πάροχος υπηρεσιών νέφους δύναται να προσφέρει υπολογιστική υποδομή, με τη μορφή ενοικιαζόμενων εικονικοποιημένων πόρων, η οποία περιλαμβάνει εικονικές μηχανές, εύρος ζώνης, δικτυακές συνδέσεις, IP διευθύνσεις, Εικονικά Τοπικά Δίκτυα (VLANs), τείχη προστασίας, εξισορροπητές φορτίου κ.α.. Ο πελάτης, αφού συνάψει κάποια συμφωνία ενοικίασης των επιθυμητών πόρων με τον πάροχο υπηρεσιών νέφους, αποκτά πρόσβαση σε αυτούς και έχει τη δυνατότητα να διαμορφώσει την IT (νεφοϋπολογιστική) υποδομή που επιθυμεί. Οι πάροχοι υπηρεσιών νέφους εγγυώνται την αξιόπιστη απόδοση και συντήρηση αυτής της υποδομής, δεσμευόμενοι από κάποια συμφωνία επιπέδου υπηρεσιών (SLA – Service Level Agreement). [4] [2]

Τα κύρια πλεονεκτήματα της IaaS υπηρεσίας, συνίσταται στις δυνατότητες επεκτασιμότητας (scalability), στην πληρωμή σύμφωνα με τη χρήση (pay as you go pricing), στη γεωγραφική ανεξαρτησία (location independence), στην φυσική προστασία των κέντρων δεδομένων και στην εξάλειψη του «μοναδικού σημείου αστοχίας» (single point of failure).

Ενδεικτικά παρατίθενται μερικά παραδείγματα από τους μεγαλύτερους IaaS παρόχους:

- Amazon EC2
- Google Compute Engine
- Microsoft Azure IaaS
- Rackspace Managed Cloud
- Alibaba Elastic Cloud Compute

1.1.2 PaaS (Platform as a Service)

Το PaaS είναι ένα μοντέλο υπηρεσίας της υπολογιστικής νέφους στο οποίο προσφέρεται, κατά απαίτηση, μια πλατφόρμα λογισμικού εφαρμογών (application software platform) από κάποιον πάροχο υπηρεσιών νέφους, με στόχο την ανάπτυξη κάποιας εφαρμογής. Πρόκειται για μια μορφή υπηρεσίας που παρέχει ένα ολοκληρωμένο περιβάλλον ανάπτυξης και εκτέλεσης εφαρμογών, το οποίο γίνεται διαθέσιμο μέσω δικτύου στους προγραμματιστές εφαρμογών.

Οι πελάτες του PaaS είναι οργανισμοί που ασχολούνται, κυρίως, με την ανάπτυξη λογισμικού. Οι προγραμματιστές χρειάζονται πλατφόρμες, όπως φυσικούς εξυπηρετητές, βάσεις δεδομένων και εξυπηρετητές δικτύου, για να εκτελούν τις εφαρμογές τους. Σε αυτό το μοντέλο, οι προγραμματιστές εφαρμογών, αναπτύσσουν με ιδιοταγείς διεπαφές προγραμματισμού εφαρμογών (APIs) την εφαρμογή τους, η οποία θα εκτελεστεί σε ένα συγκεκριμένο περιβάλλον που διαμορφώνει ο πάροχος υπηρεσιών νέφους.

Το PaaS, μεταξύ άλλων, δίνει τη δυνατότητα ενοικίασης λειτουργικών συστημάτων, προγραμματιστικών εργαλείων, χώρου αποθήκευσης και δικτυακής χωρητικότητας (network capacity) μέσω δικτύου. Στο PaaS μοντέλο, ο πελάτης αποκτά πρόσβαση σε ένα σύνολο εργαλειοθηκών (toolkits), τα οποία του επιτρέπουν να δημιουργήσει την επιθυμητή πλατφόρμα για την ανάπτυξη των εφαρμογών του. Οι PaaS πάροχοι, προσφέρουν πρόσβαση σε διάφορα σύνολα δομικών εργαλείων (building blocks) λογισμικού, επιτρέποντας την ανάπτυξη νέων εφαρμογών εύκολα και τάχιστα [7]. Ο πελάτης διαχειρίζεται πλήρως τα ανώτερα επίπεδα, περιλαμβανομένων των εφαρμογών του και ο πάροχος διαχειρίζεται τα λειτουργικά συστήματα και το υλισμικό που υποστηρίζουν την πλατφόρμα. [5]

Επιπροσθέτως, στο PaaS εμπεριέχεται, σαν υποκατηγορία, το μοντέλο υπηρεσίας CaaS (Container as a Service), το οποίο τα τελευταία χρόνια έχει πολύ μεγάλη δυναμική και απήχηση στις υπηρεσίες νέφους. Το CaaS παρέχει εργαλεία και υπηρεσίες που επιτρέπουν τη διαχείριση του κύκλου ζωής των περιεκτών, όπως την αυτόματη κατανομή, την κλιμακωσιμότητα και την ασφάλεια. Στο πλαίσιο του CaaS, οι πελάτες μπορούν να αναπτύξουν, να εκτελέσουν και να διανείμουν εφαρμογές χρησιμοποιώντας περιέκτες, χωρίς την ανάγκη να ανησυχούν για την υποκείμενη υποδομή και τις τεχνικές προδιαγραφές που, αυτή, απαιτεί. Το CaaS

αποτελεί ένα σημαντικό κομμάτι του PaaS, το οποίο παρέχει ένα ολοκληρωμένο περιβάλλον ανάπτυξης και εκτέλεσης εφαρμογών. Μέσω του CaaS, το PaaS παρέχει τη δυνατότητα διαχείρισης των περιεκτών και της υποδομής που τους υποστηρίζει, επιτρέποντας στους χρήστες να επωφεληθούν από τα πλεονεκτήματα της τεχνολογίας των περιεκτών. Αυτή η συνδυαστική προσέγγιση παρέχει την ευελιξία και την απλότητα που χρειάζονται οι επιχειρήσεις για να αναπτύξουν και να εκτελέσουν εφαρμογές με αποδοτικότητα και κλιμακωσιμότητα, απελευθερώνοντας τους από την «επίπονη» διαχείριση των υποκείμενων υποδομών. [3]

Στο παραδοσιακό μοντέλο, ο πελάτης χρειάζεται να αφιερώσει πολύ κόπο (τεχνογνωσία) και CAPEX για να χτίσει τη δική του πλατφόρμα, πάνω στην οποία θα εκτελεστούν οι εφαρμογές του και παράλληλα θα πρέπει να μεριμνά και για τη συνεχή διαχείριση της συγκεκριμένης πλατφόρμας. Στο PaaS, σε σύγκριση με το παραδοσιακό τρόπο ανάπτυξης εφαρμογών, ο πελάτης απαγκιστρώνεται από την ενασχόληση του με τα θέματα των εξυπηρετητών, των χώρων αποθήκευσης και με το εύρος ζώνης του δικτύου, που απαιτούν εξειδικευμένο προσωπικό, καθώς όλα προσφέρονται από τον πάροχο υπηρεσιών νέφους. Η προαναφερθείσα απαγκίστρωση δίνει τη δυνατότητα στον πελάτη να απασχολεί τους προγραμματιστές του αποκλειστικά για την ανάπτυξη των εφαρμογών του, αυξάνοντας την παραγωγικότητά τους, αφού δεν τους εμπλέκει σε καταστάσεις επίλυσης προβλημάτων (troubleshooting), που πιθανότατα δεν είναι μέρος του γνωστικού τους πεδίου. Ο πελάτης διαχειρίζεται πλήρως τα ανώτερα επίπεδα, περιλαμβανομένων των εφαρμογών του και ο πάροχος υπηρεσιών νέφους διαχειρίζεται τα λειτουργικά συστήματα και το υλισμικό που υποστηρίζει την πλατφόρμα.

Τα μοντέλα υπηρεσιών PaaS ενδέχεται να μην είναι μια ολοκληρωμένη λύση για κάθε εφαρμογή εντός μιας επιχείρησης. Η εκάστοτε PaaS υποδομή έχει αρκετές προκλήσεις να αντιμετωπίσει ώστε να ικανοποιήσει τις απαιτήσεις του πελάτη τόσο σε επίπεδο αυτοματισμών όσο και πολυχρηστικότητας. Επιπλέον, τα πράγματα μπορεί να είναι ακόμη πιο αποτρεπτικά σε περιπτώσεις που κάποια εφαρμογή χρειάζεται τροποποίηση για την ομαλή λειτουργία της στο παρεχόμενο υλισμικό-λογισμικό, ειδικά αν λάβουμε υπόψιν μας το κλείδωμα παρόχου (vendor lock in) το οποίο περιορίζει αισθητά την ευελιξία μετακόμισης μιας εφαρμογής από μια πλατφόρμα σε μία άλλη. [11]

Ενδεικτικά παρατίθενται μερικά παραδείγματα από τους μεγαλύτερους PaaS παρόχους:

- Microsoft Azure PaaS
- Amazon Web Services Elastic Beanstalk
- Google App Engine
- Heroku
- Oracle Cloud Platform

1.1.3 SaaS (Software as a Service)

Το SaaS μπορεί να οριστεί ως ένα μοντέλο υπηρεσίας, το οποίο παρέχει, μέσω διαδικτύου και κατά απαίτηση, ολοκληρωμένες εφαρμογές λογισμικού σε τελικούς χρήστες. Σύμφωνα με τον Paul et al [12], το SaaS μοντέλο είναι ανάλογο του μοντέλου πελάτη - εξυπηρετητή (client - server), μόνο που στην περίπτωση του SaaS η θέση του εξυπηρετητή έχει αντικατασταθεί από το κέντρο δεδομένων του παρόχου υπηρεσιών νέφους. Είναι το μοντέλο υπηρεσίας με την μεγαλύτερη απήχηση και αυτό που προσφέρθηκε πρώτο στην αγορά.

Το SaaS βρίσκεται στην κορυφή της στοίβας των μοντέλων υπηρεσιών της Υπολογιστικής Νέφους και ακολουθεί πολυχρηστική αρχιτεκτονική, με την οποία πολλοί χρήστες μοιράζονται την ίδια νεφούπολογιστική υποδομή του παρόχου. Σε αυτή την υποδομή, το λογισμικό και τα σχετικά δεδομένα φιλοξενούνται στο νέφος, στο οποίο ο χρήστης έχει πρόσβαση μέσω κάποιου διαδικτυακού φυλλομετρητή (web browser - HTTP/HTTPS). Ο πελάτης μπορεί να χρησιμοποιήσει το λογισμικό-υπηρεσία που διατίθεται από τον πάροχο υπηρεσιών νέφους, κάνοντας χρήση είτε συγκεκριμένης άδειας, είτε συνδρομής, είτε σύμφωνα με το μοντέλο “pay as you go”. Ο χρήστης έχει πρόσβαση στην εφαρμογή, αλλά δεν έχει πρόσβαση στην υποκείμενη υποδομή, όπως δίκτυα, εξυπηρετητές, λειτουργικά συστήματα και χώρους αποθήκευσης.

Στις παραδοσιακές διανομές λογισμικού, οι χρήστες πρέπει να αγοράσουν και να εγκαταστήσουν το λογισμικό στους προσωπικούς τους υπολογιστές. Στο SaaS μοντέλο, ο πάροχος υπηρεσιών νέφους ελέγχει τα διάφορα επίπεδα διαστρωμάτωσης (της υπηρεσίας) συμπεριλαμβανομένων του υλισμικού, του

λειτουργικού συστήματος, του ενδιάμεσου λογισμικού και των εφαρμογών. Ο πελάτης, η μοναδική πρόσβαση που έχει είναι στις εφαρμογές λογισμικού, τις οποίες ο πάροχος ορίζει ως διαθέσιμες προς ενοικίαση. Ο πάροχος δημιουργεί ένα στιγμιότυπο (instance) της εφαρμογής, και ακολούθως ο πελάτης συνδέεται και χρησιμοποιεί την εφαρμογή μέσω κάποιου API. Μόνο οι πάροχοι υπηρεσιών νέφους μπορούν να διαμορφώσουν, να αναβαθμίσουν και να διαχειριστούν τη λειτουργικότητα των εφαρμογών και επίσης μόνο αυτοί έχουν πρόσβαση στα αρχεία καταγραφής (log files).

Το SaaS δίνει τη δυνατότητα στις επιχειρήσεις να έχουν πρόσβαση, από παντού, σε εταιρικές υπηρεσίες μέσω του διαδικτύου, με τιμολόγηση σύμφωνα με τη χρήση, χωρίς να τους απασχολούν ζητήματα όπως η εγκατάσταση, η διαχείριση και η αναβάθμιση των εφαρμογών τους. Ο χρόνος εγκατάστασης τόσο του λογισμικού όσο και του υλισμικού στις υποδομές του πελάτη είναι μηδενικός, αφού ουσιαστικά δεν γίνεται κάποια εγκατάσταση. Το μοντέλο υπηρεσίας προσφέρει αρκετά πλεονεκτήματα για τους χρήστες. Καταρχάς, παρέχει εύκολη διαχείριση, καθώς ο πάροχος υπηρεσιών νέφους είναι υπεύθυνος για την εγκατάσταση αυτόματων διορθώσεων (patches) και ενημερώσεων (updates). Αυτό σημαίνει ότι οι χρήστες δεν χρειάζεται να ασχολούνται με τις λεπτομέρειες της συντήρησης του λογισμικού και μπορούν να επικεντρωθούν στη χρήση της εφαρμογής. [11]

Επιπλέον, το SaaS επιτρέπει την αποτελεσματική συνεργασία μεταξύ εφαρμογών λόγω της συμβατότητας των εκδόσεων. Ο πάροχος υπηρεσιών νέφους είναι υπεύθυνος για το versioning, διασφαλίζοντας ότι οι χρήστες χρησιμοποιούν την τελευταία έκδοση της εφαρμογής. Αυτό αποτρέπει τη συντήρηση παλαιωμένων εκδόσεων που μπορεί να προκαλέσουν προβλήματα ασυμβατότητας. Επιπλέον, ο πάροχος υπηρεσιών νέφους αναλαμβάνει τον έλεγχο και την παρακολούθηση (monitoring) της υποδομής, εξασφαλίζοντας την ασφάλεια και τη διαθεσιμότητα της.

Συνολικά, το μοντέλο υπηρεσίας SaaS προσφέρει την ευκολία, την ασφάλεια και τη συντήρηση της εφαρμογής λογισμικού μέσω του παρόχου υπηρεσιών νέφους, επιτρέποντας στους χρήστες να απολαύσουν τις πλήρεις λειτουργικότητες της εφαρμογής, χωρίς πρόσθετες διαχειριστικές επιβαρύνσεις. [7]

Ενδεικτικά παρατίθενται μερικά παραδείγματα από τους μεγαλύτερους SaaS παρόχους:

- Microsoft 365
- Salesforce CRM
- Adobe Creative Cloud
- Zoom
- Netflix

1.1.4 XaaS (Anything as a Service)

Χάρη στην αλματώδη ανάπτυξη των μοντέλων υπηρεσιών της υπολογιστικής νέφους και των διάφορων υπηρεσιών ή πόρων που προσφέρονται μέσω αυτών, έχει δημιουργηθεί η έννοια XaaS (Anything as a Service). Οι υπηρεσίες XaaS περιλαμβάνουν όλες τις πιθανές κατηγορίες υπηρεσιών οι οποίες μπορούν να προσφερθούν από τους παρόχους, προεκτείνοντας τις υπηρεσίες που ορίζονται από το Μοντέλο του NIST, όπως DBaaS, που προσφέρει υπηρεσίες βάσης δεδομένων, CaaS, που προσφέρει υπηρεσίες περιεκτών, DaaS που προσφέρει υπηρεσίες εικονικής επιφάνειας εργασίας, SECaaS, που προσφέρει ασφάλεια ως υπηρεσία, STaaS, που προσφέρει τη δυνατότητα αποθήκευσης δεδομένων σε διάφορες μορφές (files, blocks, images, objects) κ.α.. [4]

1.2 Μοντέλα Ανάπτυξης Υπολογιστικού Νέφους

Στο NIST μοντέλο, ορίζονται τα ακόλουθα τέσσερα μοντέλα ανάπτυξης της υπολογιστικής νέφους:

- Ιδιωτικό Νέφος (Private Cloud)
- Νέφος Κοινότητας (Community Cloud)
- Δημόσιο Νέφος (Public Cloud)
- Υβριδικό Νέφος (Hybrid Cloud)

Τα μοντέλα ανάπτυξης της υπολογιστικής νέφους παρέχουν μια περιγραφή του τρόπου λειτουργίας του νέφους και καθορίζουν ποιος έχει πρόσβαση στους πόρους της υπηρεσίας. Στο ιδιωτικό νέφος, η υποδομή νέφους παρέχεται για αποκλειστική χρήση από έναν μόνο οργανισμό. Αντίθετα, στο δημόσιο νέφος, οι πόροι είναι διαθέσιμοι για ανοιχτή χρήση από το ευρύ κοινό. Το νέφος κοινότητας παρέχει

υπηρεσίες για αποκλειστική χρήση από διάφορους οργανισμούς με κοινά ενδιαφέροντα και ανάγκες. Τέλος, το υβριδικό νέφος αποτελείται από μία σύνθεση δύο ή περισσότερων διαφορετικών μοντέλων ανάπτυξης (ιδιωτικό, δημόσιο, κοινότητας), τα οποία παραμένουν αυτόνομα αλλά συνδέονται μεταξύ τους μέσω μιας τυποποιημένης ή ιδιοταγούς τεχνολογίας. Αυτή η τεχνολογία επιτρέπει τη φορητότητα δεδομένων και εφαρμογών μεταξύ των διάφορων μοντέλων. [13]

Ακολουθεί μία πιο μεστή περιγραφή των τεσσάρων αυτών μοντέλων ανάπτυξης.



Εικόνα 2. Μοντέλα ανάπτυξης υπολογιστικής νέφους [14]

1.2.1 Ιδιωτικό Νέφος (Private Cloud)

Το ιδιωτικό νέφος φιλοξενείται σε φυσικές υλικές υποδομές, οι οποίες χρησιμοποιούνται αποκλειστικά από έναν οργανισμό. Οι υλικές αυτές υποδομές μπορούν να βρίσκονται είτε σε κέντρο δεδομένων κάποιου παρόχου υπηρεσιών νέφους, είτε σε ιδιόκτητο κέντρο δεδομένων του ίδιου του οργανισμού που το χρησιμοποιεί. Η βασική διαφορά μεταξύ ενός ιδιωτικού νέφους και άλλων μορφών νέφους είναι ότι κανένας άλλος, πέραν του οργανισμού στον οποίο ανήκει το νέφος, δεν επιτρέπεται να χρησιμοποιεί τους υλικούς πόρους για το δικό του νέφος, ανεξαρτήτως του που αυτό βρίσκεται γεωχωρικά. Το κύριο κίνητρο για την υλοποίηση ενός ιδιωτικού νέφους στις υποδομές του ίδιου του οργανισμού είναι, συνήθως, η ανάγκη φύλαξης ευαίσθητων δεδομένων στις εγκαταστάσεις του. [15]

Όταν ένας οργανισμός δημιουργεί και αναπτύσσει σύνθετες υπηρεσίες, η δημιουργία ενός ιδιωτικού νέφους, που υποστηρίζει τις εσωτερικές ανάγκες του, αποτελεί μία ιδιαίτερα πρακτική λύση με υψηλό βαθμό αυτοματοποίησης. Μέσω του ιδιωτικού νέφους, ο οργανισμός έχει πλήρη έλεγχο και διαχείριση των υπηρεσιών του, ενώ ταυτόχρονα διασφαλίζει την αποκλειστική χρήση των υλικών πόρων από τον ίδιο τον οργανισμό. Επιπλέον, σε ορισμένες περιπτώσεις, ένας οργανισμός

μπορεί να επιθυμεί να δημιουργήσει ένα περιβάλλον νέφους που βρίσκεται πίσω από το τείχος προστασίας του ίδιου του οργανισμού. Με αυτόν τον τρόπο, οι ευαίσθητες πληροφορίες και τα δεδομένα του οργανισμού παραμένουν απομονωμένα και προστατευμένα από εξωτερικές απειλές. Η ενσωμάτωση μιας λύσης ιδιωτικού νέφους πίσω από ένα τείχος προστασίας παρέχει ένα επιπλέον επίπεδο ασφάλειας και ελέγχου για τον οργανισμό, επιτρέποντάς του να διαχειρίζεται τα δεδομένα του με πιο αποτελεσματικό τρόπο.

Σε αντίθεση με ένα παραδοσιακό ιδιωτικό κέντρο δεδομένων, το ιδιωτικό νέφος παρέχει μια κοινή δεξαμενή πόρων, που έχουν βελτιστοποιηθεί για τη χρήση από το υπολογιστικό τμήμα του οργανισμού. Συνήθως, τα σύγχρονα ιδιωτικά κέντρα δεδομένων χρησιμοποιούν τεχνολογίες εικονικοποίησης. Παρότι η εικονικοποίηση παρέχει κάποιο επίπεδο ευελιξίας στον τρόπο με τον οποίο το φυσικό υλικό αλληλοεπιδρά με τις προσφερόμενες υπηρεσίες, η διαμόρφωση αυτών των υπηρεσιών εξαρτάται στενά από το υποκείμενο φυσικό υλικό. Το νέφος προσφέρει ένα επιπλέον επίπεδο αφαίρεσης από το φυσικό υλικό και επαφίεται περισσότερο σε μία μορφή επικοινωνίας βασισμένη σε κάποια διεπαφή προγραμματισμού υπηρεσίας (API) η οποία είναι διαχειρίσιμη μέσω κάποιου λογισμικού εικονικοποίησης. Έτσι, οποιοδήποτε επίπεδο υπηρεσίας απαιτείται για τον οργανισμό μπορεί να προσφερθεί από το ιδιωτικό νέφος.

Με τη χρήση ενός ιδιωτικού νέφους, ο οργανισμός έχει τη δυνατότητα να αξιοποιήσει αποδοτικά τους διαθέσιμους πόρους, καθώς οι προσαρμοσμένες υπηρεσίες μπορούν να παρέχονται με βάση τις ανάγκες του. Η επικοινωνία με το ιδιωτικό νέφος γίνεται μέσω προγραμματιστικών διεπαφών (API), που επιτρέπουν στον οργανισμό να αναπτύσσει, να διαμορφώνει και να διαχειρίζεται τις εφαρμογές και τις υπηρεσίες του με ευελιξία και αποτελεσματικότητα. Αυτό επιτρέπει στον οργανισμό να προσαρμόζεται γρήγορα σε μεταβαλλόμενες ανάγκες και να αναπτύσσει καινοτόμες λύσεις με μεγαλύτερη ευελιξία και χαμηλότερο κόστος. [4]

Συνολικά, το ιδιωτικό νέφος παρέχει στον οργανισμό πλήρη έλεγχο, ασφάλεια και ευελιξία για την υποστήριξη των εσωτερικών του αναγκών. Ο οργανισμός μπορεί να προσαρμόσει και να διαμορφώσει το ιδιωτικό νέφος ακριβώς όπως το επιθυμεί, επιτρέποντάς του να επιτύχει τις ειδικές απαιτήσεις και προδιαγραφές του.

1.2.2 Δημόσιο Νέφος (Public Cloud)

Τα δημόσια νέφη είναι ίσως το πιο γνωστό από τα μοντέλα ανάπτυξης νέφους, κυρίως επειδή ήταν και τα πρώτα βιώσιμα περιβάλλοντα υπολογιστικής νέφους για τους οργανισμούς. Κυρίως λόγω της ποικιλίας των χρηστών, τα δημόσια νέφη προσφέρουν μια ευρεία γκάμα από υπολογιστικές υπηρεσίες. Από SaaS μέχρι IaaS τα δημόσια νέφη είναι δομημένα έτσι ώστε να προσπαθήσουν να ικανοποιήσουν την οποιαδήποτε επιθυμία των εκάστοτε χρηστών. Όλοι οι μεγάλοι πάροχοι υπηρεσιών νέφους συνήθως προσφέρουν IaaS, PaaS και SaaS δυνατότητες ανάλογα με τις απαιτήσεις των χρηστών.

Τα δημόσια νέφη, είναι μια μορφή υποδομής υπολογιστικού νέφους, που φιλοξενείται σε ειδικά σχεδιασμένες φυσικές εγκαταστάσεις - κέντρα δεδομένων - του παρόχου της υπηρεσίας. Αυτά τα κέντρα δεδομένων είναι συχνά εξοπλισμένα με προηγμένες τεχνολογίες για να προσφέρουν υψηλό επίπεδο ασφάλειας, αποδοτικότητας και ευελιξίας. Οι πάροχοι υπηρεσιών νέφους μπορούν να είναι διάφοροι οργανισμοί, από επαγγελματικές εταιρείες και ακαδημαϊκούς οργανισμούς μέχρι κυβερνητικές οντότητες. Αυτοί οι πάροχοι προσφέρουν μια ποικιλία υπηρεσιών, μερικές φορές χωρίς κόστος, αλλά συνήθως με βάση κάποιο μοντέλο χρέωσης. Αυτές οι υπηρεσίες μπορούν να περιλαμβάνουν χώρους αποθήκευσης δεδομένων, υπολογιστική ισχύ και διάφορες δικτυακές λειτουργίες. [15]

Το βασικό πλεονέκτημα των δημόσιων νεφών είναι η προσβασιμότητα. Οποιοσδήποτε με πρόσβαση στο διαδίκτυο μπορεί να χρησιμοποιήσει τις υπηρεσίες που παρέχουν. Ωστόσο, η ασφάλεια των δεδομένων και η προστασία της ιδιωτικότητας είναι συχνά πηγές ανησυχίας, καθώς τα δεδομένα μπορεί να είναι προσβάσιμα σε άλλους πελάτες του ίδιου παρόχου. Επίσης, εάν ο πάροχος υπολογιστικού νέφους αντιμετωπίζει τεχνικά προβλήματα, αυτό μπορεί να επηρεάσει όλους τους πελάτες του.

Στο δημόσιο νέφος, ο πάροχος υπηρεσιών νέφους αναλαμβάνει την αποκλειστική ευθύνη για τη διαχείριση της φυσικής υλικής υποδομής, όπου οι χρήστες δεν έχουν απευθείας πρόσβαση. Αυτό σημαίνει ότι οι χρήστες εξαρτώνται εξ' ολοκλήρου από τις πολιτικές διαχείρισης, ασφάλειας και λειτουργίας που ορίζει ο πάροχος, καθώς και από τη διαθεσιμότητα και ασφάλεια των δεδομένων και των πόρων τους που φιλοξενούνται στο δημόσιο νέφος. Ωστόσο, αυτό δεν σημαίνει ότι

οι χρήστες δεν φέρουν ευθύνη για την ασφάλεια των δεδομένων τους. Πρέπει να λαμβάνουν υπόψη τους τις βέλτιστες πρακτικές και να εφαρμόζουν αποτελεσματικά μέτρα ασφαλείας για να προστατεύσουν τα δεδομένα τους από απειλές όπως η απώλεια, η διαρροή ή η μη εξουσιοδοτημένη πρόσβαση. Επιπλέον, οι χρήστες πρέπει να είναι ενήμεροι για τις πολιτικές ασφαλείας που έχουν θεσπιστεί από τον πάροχο υπηρεσιών νέφους και να τις ακολουθούν προσεκτικά για να διασφαλίσουν την προστασία των δεδομένων τους. [11]

Επίσης, είναι σημαντικό να σημειωθεί ότι ο πάροχος υπηρεσιών νέφους δεν εγγυάται απαραίτητα ικανοποιητικά επίπεδα διασφάλισης των υποδομών του. Οι χρήστες πρέπει να εξετάζουν προσεκτικά τις πολιτικές ασφαλείας και τις πρακτικές που εφαρμόζονται από τον πάροχο υπηρεσιών νέφους και να αξιολογούν την απόδοση και την αξιοπιστία του, προτού εμπιστευθούν τα δεδομένα τους σε αυτόν. Η διατήρηση ανοικτής επικοινωνίας με τον πάροχο και η επαγρύπνηση για τυχόν ανησυχητικές ενδείξεις ή προβλήματα ασφαλείας είναι σημαντικά για τη διασφάλιση της ακεραιότητας και της ασφαλείας των δεδομένων των χρηστών.

1.2.3 Νέφος Κοινότητας (Community Cloud)

Το νέφος κοινότητας είναι μία ειδική μορφή ιδιωτικού νέφους που διαφέρει από το γεγονός ότι παρέχει πρόσβαση όχι μόνο σε έναν οργανισμό, αλλά σε περισσότερους από έναν, ενώ παράλληλα δεν είναι προσβάσιμο από το ευρύ κοινό. Αυτό σημαίνει ότι μόνο τα μέλη της κοινότητας έχουν πρόσβαση στο νέφος. Το νέφος κοινότητας μπορεί να είναι διαχειρίσιμο είτε από τους ίδιους τους οργανισμούς που συμμετέχουν, είτε από έναν εξωτερικό πάροχο υπηρεσιών νέφους που παρέχει υπηρεσίες διαχείρισης. Επίσης, το νέφος κοινότητας μπορεί να φιλοξενείται είτε σε ιδιόκτητες εγκαταστάσεις των μελών της κοινότητας, είτε στις εγκαταστάσεις που παρέχονται από τον εξωτερικό πάροχο υπηρεσιών νέφους. [11]

Η δυνατότητα διαχείρισης του νέφους κοινότητας από τους ίδιους τους οργανισμούς παρέχει αυτονομία και ελευθερία στη λειτουργία τους, ενώ η εξωτερική διαχείριση από έναν πάροχο υπηρεσιών νέφους μπορεί να παρέχει ειδικούς επαγγελματικούς πόρους και τεχνογνωσία για την αποτελεσματική διαχείριση και ασφάλεια του νέφους κοινότητας. Ανεξαρτήτως του τρόπου διαχείρισης και φιλοξενίας, το νέφος κοινότητας προσφέρει τη δυνατότητα συνεργασίας και

ανταλλαγής πόρων μεταξύ των μελών της κοινότητας, επιτρέποντας την αποτελεσματική και ασφαλή διαχείριση των δεδομένων και των εφαρμογών τους.

Η ανάγκη για τη δημιουργία ενός νέφους κοινότητας απορρέει από την ύπαρξη μίας ομάδας οργανισμών με κοινά ενδιαφέροντα και συμφέροντα. Σε αυτήν την περίπτωση, ένα κοινοτικό νέφος μπορεί να αναδύεται για να ανταποκριθεί σε απαιτήσεις όπως κοινές αποστολές, απαιτήσεις ασφάλειας ή πολιτικές διατάξεις. Ένα παράδειγμα κοινοτικού νέφους μπορεί να είναι μία κοινοπραξία διαφόρων νοσοκομείων που συνεργάζονται για να μοιραστούν την πρόσβαση στα ιστορικά των ασθενών τους. Στο νέφος αποθηκεύεται η βάση δεδομένων με τα ιατρικά αρχεία των ασθενών, η οποία είναι προσβάσιμη από τα νοσοκομεία που συμμετέχουν στην κοινότητα. Ωστόσο, η πρόσβαση αυτή δεν είναι διαθέσιμη σε άλλους οργανισμούς ή στο ευρύ κοινό, επιτρέποντας έτσι τη διατήρηση της ιδιωτικότητας και της ασφάλειας των δεδομένων των ασθενών.

Μέσω του κοινοτικού νέφους, τα νοσοκομεία μπορούν να συνεργαστούν, να ανταλλάσσουν πληροφορίες και να διασφαλίσουν τη συνοχή και τη συνέπεια στην παροχή υπηρεσιών υγείας. Παράλληλα, η αποκλειστική πρόσβαση στα δεδομένα εξασφαλίζει ότι οι πληροφορίες των ασθενών παραμένουν ασφαλείς και εμπιστευτικές, προστατεύοντας την ιδιωτικότητα τους και συμμορφούμενοι με τις νομικές απαιτήσεις περί προστασίας δεδομένων. [4]

Αυτό το μοντέλο ανάπτυξης είναι περισσότερο μία θεωρητική πιθανότητα, καθώς στην πράξη δεν έχει υπάρξει μεγάλο ενδιαφέρον σε επίπεδο υλοποίησης. [10]

1.2.4 Υβριδικό Νέφος (Hybrid Cloud)

Το υβριδικό νέφος αντιπροσωπεύει μία ενδιαφέρουσα προσέγγιση, καθώς συνδυάζει τα πλεονεκτήματα και την ευελιξία του δημόσιου και του ιδιωτικού νέφους από έναν οργανισμό. Το δημόσιο νέφος παρέχει ευρεία προσβασιμότητα μέσω του διαδικτύου, ενώ το ιδιωτικό νέφος προσφέρει πλήρη έλεγχο και αποκλειστική πρόσβαση στους ιδιοκτήτες του.

Το υβριδικό νέφος επιτρέπει την ομαλή ενσωμάτωση και διασύνδεση υπηρεσιών που βρίσκονται σε δημόσια, ιδιωτικά και κοινοτικά νέφη, με στόχο τη δημιουργία ενός ενιαίου εικονικού υπολογιστικού περιβάλλοντος. Αυτή η ολοκλήρωση επιτρέπει στις υποδομές και τις υπηρεσίες να αλληλοεπιδρούν μεταξύ τους με έναν ευθύ και διαφανή τρόπο. Το υβριδικό νέφος μπορεί να χαρακτηριστεί ως μία ρευστή

μίξη από φυσικές και εικονικές υποδομές, οι οποίες μπορούν να βρίσκονται είτε τοπικά είτε απομακρυσμένα.

Η υιοθέτηση ενός υβριδικού νέφους μπορεί να προσφέρει πλεονεκτήματα όπως ευελιξία στην ανάπτυξη και στην επέκταση των υπηρεσιών, δυνατότητα εκμετάλλευσης της υπάρχουσας υποδομής και εξοικονόμηση κόστους. Ωστόσο, πρέπει να ληφθούν υπόψη η διαχείριση της ασφάλειας, η επικοινωνία μεταξύ των διαφορετικών περιβαλλόντων και η πολυπλοκότητα της υποδομής για επιτυχή υλοποίηση και λειτουργία του υβριδικού νέφους. Αυτό το μοντέλο υλοποίησης θα μπορούσε να εμφανιστεί σαν μια ενδιάμεση κατάσταση της μεταφοράς υπηρεσιών στο νέφος από κάποιον οργανισμό ή και σαν μια αέναη υβριδική διάταξη. Πολλοί οργανισμοί υποστηρίζουν τη μεταφορά των υπηρεσιών τους στο νέφος και αυτή η μεταφορά συνήθως πραγματοποιείται σε φάσεις, σταδιακά. Σε αυτήν τη σταδιακή διαδικασία, υλοποιείται ένα υβριδικό νέφος, όπου ορισμένες υπηρεσίες βρίσκονται στο νέφος ενώ άλλες παραμένουν στο παραδοσιακό κέντρο δεδομένων του οργανισμού.

Στον αντίποδα, ορισμένοι οργανισμοί κρίνουν ότι δεν είναι όλες οι υπηρεσίες τους κατάλληλες για μεταφορά στο νέφος. Σε αυτές τις περιπτώσεις, πραγματοποιείται μερική μεταφορά στο νέφος, με τον σκοπό να διατηρηθούν ορισμένες υπηρεσίες στο παραδοσιακό κέντρο δεδομένων τους. Παράλληλα, οι πόροι που υπάρχουν στα διάφορα περιβάλλοντα ενσωματώνονται και επικοινωνούν άμεσα μεταξύ τους. Ένα παράδειγμα τέτοιου είδους επικοινωνίας θα μπορούσε να είναι η ύπαρξη μιας βάσης δεδομένων σε ένα ιδιωτικό κέντρο δεδομένων που επικοινωνεί με έναν εξυπηρετητή σε ένα δημόσιο νέφος. Μέσω αυτής της επικοινωνίας, οι υπηρεσίες μπορούν να συνεργάζονται μεταξύ των δύο περιβαλλόντων, επιτρέποντας την αξιοποίηση των πλεονεκτημάτων του κάθε περιβάλλοντος και τη διατήρηση της ευελιξίας και της ασφάλειας των δεδομένων.

Στην πραγματικότητα, το υβριδικό μοντέλο υλοποίησης νέφους αποτελεί την πιο διαδεδομένη πρακτική, καθώς ένας σημαντικός αριθμός οργανισμών δεν είναι απολύτως έτοιμος ή ικανός να μεταφέρει όλες τις υπηρεσίες τους στο νέφος, ακόμη και αν αυτό αφορά αποκλειστικά θέματα ασφάλειας [10]. Οι λόγοι που καθιστούν το υβριδικό μοντέλο προτιμητέο είναι πολλοί. Η μετάβαση όλων των υπηρεσιών στο νέφος μπορεί να απαιτεί σημαντικές αλλαγές στην υπάρχουσα υποδομή και πολύ υψηλό κόστος, ενώ ορισμένες ευαίσθητες υπηρεσίες μπορεί να απαιτούν επιπλέον

επίπεδα ασφάλειας που δεν μπορούν να προσφερθούν εξολοκλήρου από το νέφος. Ορισμένοι οργανισμοί μπορεί να επιθυμούν να διατηρήσουν τον πλήρη έλεγχο και την πρόσβαση σε κρίσιμες υπηρεσίες που εκτελούνται στο εσωτερικό τους δίκτυο, ενώ ταυτόχρονα επωφελούνται από τα πλεονεκτήματα της ευελιξίας και της κλιμακωσιμότητας που προσφέρει το νέφος. [4]

Έτσι, το υβριδικό μοντέλο επιτρέπει την επίτευξη μιας ισορροπημένης λύσης, όπου οι οργανισμοί μπορούν να διατηρήσουν τον απαραίτητο έλεγχο και ασφάλεια για ορισμένες υπηρεσίες, ενώ παράλληλα αξιοποιούν τα οφέλη της ευελιξίας, της αποθήκευσης και της επεκτασιμότητας που προσφέρει το νέφος.

ΚΕΦΑΛΑΙΟ 2

Τρόποι Διάθεσης Υπολογιστικών Πόρων στο Νέφος

Τα τελευταία χρόνια, η πρόοδος τόσο στις τεχνολογίες υλισμικού όσο και λογισμικού έχει επιφέρει σημαντικές αλλαγές στον τρόπο λειτουργίας και παροχής εφαρμογών και υπηρεσιών. Οι χρήστες έχουν γίνει ακόμα πιο απαιτητικοί, αναζητώντας προηγμένες λειτουργίες, ασφάλεια, αποδοτικότητα και αξιοπιστία. Ταυτόχρονα, οι επιχειρήσεις αντιμετωπίζουν αυξημένες ανάγκες στους τομείς των υποδομών και των υπηρεσιών, επηρεαζόμενες από την αύξηση των δεδομένων και την ανάγκη για συνεχή διαθεσιμότητα και ευελιξία.

Οι παραπάνω ανάγκες σε συνδυασμό με την ευκολότερη και πιο οικονομική πρόσβαση στο διαδίκτυο έχουν αυξήσει τις απαιτήσεις αναφορικά με τη βέλτιστη αξιοποίηση των φυσικών υλικών πόρων, από τους παρόχους υπηρεσιών και εφαρμογών. Οι πάροχοι, συμπεριλαμβανομένων των παρόχων υπηρεσιών νέφους, αντιμετωπίζουν την πρόκληση να αξιοποιήσουν αποτελεσματικά τους πόρους τους, με στόχο τη μείωση του κεφαλαίου επένδυσης (CAPEX) και του λειτουργικού κόστους (OPEX). Κατά το παρελθόν, διάφορες τεχνολογίες έχουν προσπαθήσει να δώσουν μια απάντηση σε αυτές τις προκλήσεις και στον τρόπο με τον οποίο θα διατίθενται οι υπολογιστικοί πόροι. Στις μέρες μας, υπάρχουν τρεις βασικοί τρόποι με τους οποίους μπορούν να διατεθούν οι υπολογιστικοί πόροι στους οργανισμούς.

Ο πρώτος τρόπος είναι η πρόσβαση σε ένα πλήρες φυσικό μηχάνημα, γνωστό ως Εξυπηρετητής Γυμνού Μετάλλου (Bare Metal Server). Αυτή η προσέγγιση παρέχει την μεγαλύτερη ευελιξία και τον πληρέστερο έλεγχο των πόρων, αλλά συχνά οδηγεί σε λιγότερο αποτελεσματική αξιοποίηση των πόρων, καθώς κάθε μηχάνημα αφιερώνεται αποκλειστικά σε έναν χρήστη.

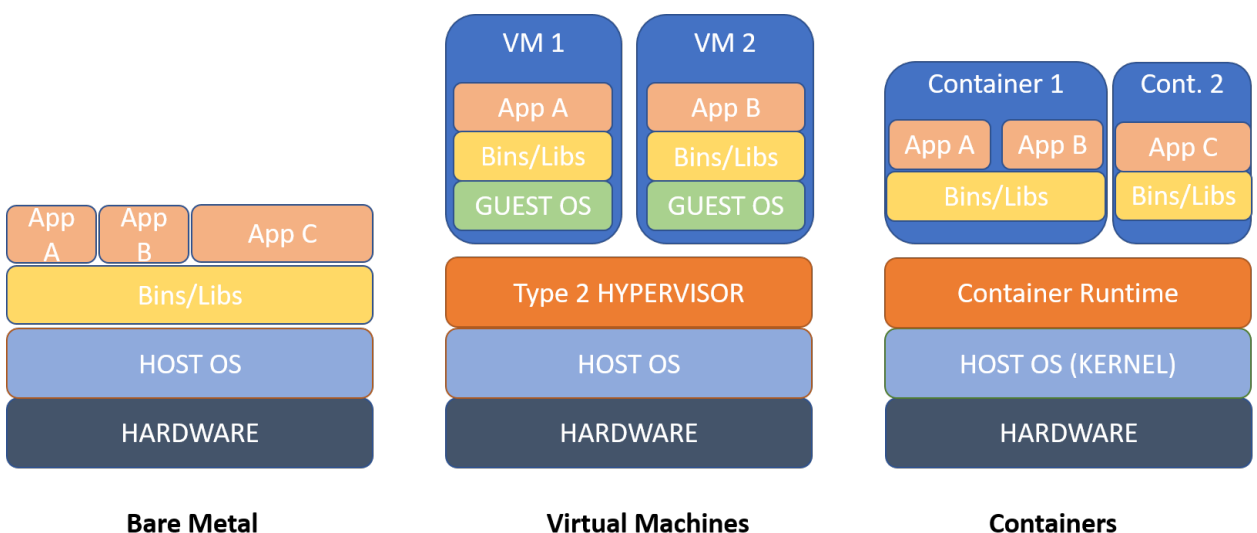
Ο δεύτερος τρόπος είναι η χρήση Εικονικών Μηχανών (Virtual Machines). Οι εικονικές μηχανές επιτρέπουν τη δημιουργία πολλαπλών εικονικών περιβαλλόντων σε ένα φυσικό μηχάνημα, επιτρέποντας την καλύτερη αξιοποίηση των πόρων. Κάθε εικονική μηχανή λειτουργεί ανεξάρτητα, με το δικό της λειτουργικό σύστημα και τις

δικές της εφαρμογές, ενώ παράλληλα μοιράζεται, με άλλες εικονικές μηχανές, τους πόρους του φυσικού μηχανήματος, στο οποίο εκτελείται.

Τέλος, ο τρίτος τρόπος είναι η χρήση εικονικών υπηρεσιών, γνωστών και ως «Περιέκτες» (Containers). Οι περιέκτες προσφέρουν ακόμα μεγαλύτερη ευελιξία και αποτελεσματική αξιοποίηση των πόρων σε σχέση με τις εικονικές μηχανές. Αναπτύσσονται ως ανεξάρτητα περιβάλλοντα, αλλά χρησιμοποιούν το ίδιο λειτουργικό σύστημα με τον φυσικό υπολογιστή, καθιστώντας τους πιο ελαφριούς και γρήγορους. Οι περιέκτες επίσης μοιράζονται τους πόρους του φυσικού μηχανήματος, επιτρέποντας την ταυτόχρονη εκτέλεση πολλαπλών εφαρμογών σε έναν υπολογιστή.

Η πιο συνοπτική προσέγγιση, σε αυτό που διαφοροποιεί τις εικονικές μηχανές από τους περιέκτες, είναι το επίπεδο στο οποίο γίνεται η αφαίρεση (abstraction). Οι εικονικές μηχανές και οι υπερεπτόπτες «αφαιρούν» το υλικό και επιτρέπουν την εκτέλεση λειτουργικών συστημάτων. Οι περιέκτες - τεχνικά μηχανές περιεκτών - αφαιρούν το λειτουργικό σύστημα και επιτρέπουν την εκτέλεση εφαρμογών. [16]

Με τους παραπάνω τρόπους, οι οργανισμοί / χρήστες έχουν τη δυνατότητα να επιλέξουν τον κατάλληλο τρόπο αξιοποίησης των υπολογιστικών πόρων, λαμβάνοντας υπόψη τις απαιτήσεις τους για απόδοση, ασφάλεια και οικονομία. [10]



Σχήμα 1. Τρόποι διάθεσης πόρων

2.1 Εξυπηρετητές Γυμνού Μετάλλου (Bare-Metal Servers)

Ένας εξυπηρετητής γυμνού μετάλλου είναι μια φυσική μηχανή που μισθώνεται από κάποιον οργανισμό και ομοιάζει πιο πολύ στο παραδοσιακό μοντέλο φιλοξενίας που προσφέρουν τα κέντρα δεδομένων. Ουσιαστικά, παραβιάζει τον ορισμό του NIST για την υπολογιστική νέφους, αφού δεν ικανοποιεί την απαίτηση για ενοποιημένους πόρους. Παρόλα αυτά, λογίζεται ως μία υπηρεσία νέφους, αφού είναι δυνατή η προμήθεια των εξυπηρετητών γυμνού μετάλλου, από τους οργανισμούς, με τρόπο κατά-απαίτησης αυτό-εξυπηρέτησης.

Προσφέρεται ως δυνατότητα από όλους τους μεγάλους παρόχους υπηρεσιών νέφους και παρέχει στον πελάτη άμεση πρόσβαση στον επεξεργαστή, τη μνήμη και τον χώρο αποθήκευσης, στον οποίο μπορεί να εγκατασταθεί ένα λειτουργικό σύστημα. Ένας εξυπηρετητής γυμνού μετάλλου επιτρέπει σε μεγάλο βαθμό τον έλεγχο της διαμόρφωσης της φυσικής μηχανής, αξιοποιώντας πλήρως το δυναμικό των υλικών πόρων. Κατά συνέπεια, χρησιμοποιείται για φόρτους εργασίας που χρειάζονται βαθιά πρόσβαση σε λειτουργίες υλικού ή λογισμικού που δεν μπορούν να προσφερθούν από τους εικονικούς διακομιστές. Αυτές μπορεί να είναι είτε υπηρεσίες παλαιού τύπου (legacy), είτε εξειδικευμένοι προσαρμοσμένοι φόρτοι εργασίας που απαιτούν την πλήρη απόδοση του υλικού, είτε εξειδικευμένες προσαρμοσμένες εφαρμογές που απαιτούν εξειδικευμένη υποστήριξη.

Ο εξυπηρετητής γυμνού μετάλλου αποτελεί μια επιλογή που συμπληρώνει τις εικονικές υπηρεσίες, προσφέροντας μια ευέλικτη και εξειδικευμένη λύση για τους πελάτες που έχουν συγκεκριμένες απαιτήσεις υλικού και απόδοσης.

2.2 Εικονικοποίηση

Η εικονικοποίηση είναι μια θεμέλιος τεχνολογία, που έκανε δυνατή την εγκαθίδρυση και την ευρεία απήχηση της υπολογιστικής νέφους. Ο όρος εικονικοποίηση αναφέρεται στην αφαίρεση των υπολογιστικών πόρων (κεντρική μονάδα επεξεργασίας, χώρος αποθήκευσης, δίκτυο, μνήμη, στοίβα υπηρεσιών και βάσεις δεδομένων κ.α.) από το υλισμικό και του διαχωρισμού των πόρων αυτών σε πολλούς εικονικοποιημένους πόρους, οι οποίοι με τη σειρά τους μπορούν να ανατεθούν σε διαφορετικές μηχανές. Το λογισμικό που υλοποιεί την αφαίρεση του υλισμικού, ώστε να μπορεί να εικονικοποιηθεί, είναι ο υπερεπόπτης (Hypervisor) [17].

Οι τεχνολογίες εικονικοποίησης δίνουν τη δυνατότητα νεφικών πολυχρηστικών επαγγελματικών μοντέλων, αφού δύνανται να προσφέρουν κλιμακώσιμες πλατφόρμες, από διαμοιραζόμενους πόρους, σε όλα τα είδη χρηστών [18] [19].

2.2.1 Είδη εικονικοποίησης

Η εικονικοποίηση επιφέρει ενοποίηση των κέντρων δεδομένων και βελτιστοποιημένη λειτουργική αποτελεσματικότητα στον τομέα της πληροφορικής. Οι σύγχρονοι οργανισμοί έχουν υιοθετήσει μία ή περισσότερες τεχνολογίες εικονικοποίησης στα κέντρα δεδομένων τους. Αυτές περιλαμβάνουν την εικονικοποίηση βασισμένη στο υλικό, όπως το KVM, το Xen, το VirtualBox κ.α., την εικονικοποίηση βασισμένη στο λειτουργικό σύστημα, όπως το Docker, το LXD, το containerd κ.α., την εικονικοποίηση αποθηκευτικών χώρων όπως τα NAS και SAN, την εικονικοποίηση βάσεων δεδομένων, καθώς και την εικονικοποίηση εφαρμογών ή λογισμικού όπως ο Apache Tomcat, το JBoss και το WebSphere, μεταξύ άλλων.

Οι τεχνολογίες εικονικοποίησης προσφέρουν ευελιξία, απομόνωση και αποδοτική διαχείριση των πόρων, καθιστώντας δυνατή τη δημιουργία ευέλικτων και κλιμακούμενων υποδομών. Οι χρήστες μπορούν να αξιοποιήσουν τους εικονικούς πόρους για την εκτέλεση εφαρμογών και υπηρεσιών, εκμεταλλευόμενοι την δυνατότητα για ασφαλή απομακρυσμένη πρόσβαση στα προσφερόμενα απομονωμένα περιβάλλοντα. Επιπλέον, η εικονικοποίηση επιτρέπει την ευέλικτη και αποτελεσματική διαχείριση των εφαρμογών και των υποδομών, μειώνοντας τον χρόνο και το κόστος ανάπτυξης, αναβάθμισης και συντήρησης των συστημάτων [20].

Επομένως, η εικονικοποίηση αποτελεί ένα ισχυρό εργαλείο για τη δημιουργία ευέλικτων, αποδοτικών και ασφαλών υπολογιστικών περιβαλλόντων στην πληροφορική [18]. Στη συνέχεια αναλύονται δύο βασικά είδη εικονικοποίησης της υπολογιστικής νέφους:

- Εικονικοποίηση βασισμένη στο Υλισμικό (Εικονικές Μηχανές)
- Εικονικοποίηση βασισμένη στο Λειτουργικό Σύστημα (Περιέκτες)

2.3 Εικονικές Μηχανές

Οι Εικονικές Μηχανές αποτελούν έναν από τους πιο συνήθεις τρόπους προσφοράς υπολογιστικής ισχύος από τους παρόχους υποδομής νέφους. Οι εικονικές μηχανές εκτελούνται στο υλισμικό κάποιου φυσικού εξυπηρετητή (χρήση εικονικών επεξεργαστών, μνήμης, αποθήκευσης και δικτύου) και μπορούν να μεταφερθούν, ακόμα και κατά τη διάρκεια λειτουργίας τους, από έναν εξυπηρετητή σε κάποιον άλλο, εφόσον αυτός πληροί τις απαιτήσεις για φυσικούς πόρους των εικονικών μηχανών.

Οι οργανισμοί έχουν πρόσβαση στο λειτουργικό σύστημα των εικονικών μηχανών, όπου μπορούν να εγκαθιστούν και να εκτελούν διάφορες εφαρμογές και προγράμματα. Συνήθως, οι οργανισμοί εκτελούν εικονικές μηχανές στα δικά τους κέντρα δεδομένων, οπότε η διαδικασία εγκατάστασης εικονικών μηχανών στο νέφος είναι μια οικεία διαδικασία, καθώς είναι παρόμοια. Επιπλέον, είναι δυνατή η δημιουργία προτύπων (templates) από ήδη υλοποιημένες εικονικές μηχανές, τα οποία μπορούν εύκολα και γρήγορα να επαναχρησιμοποιηθούν και να προσαρμοστούν στις ανάγκες του οργανισμού.

Ωστόσο, οι εικονικές μηχανές δεν χρησιμοποιούνται όταν απαιτείται έλεγχος των βαθύτερων επιπέδων του υλισμικού, όπου προτιμώνται οι εξυπηρετητές γυμνού μετάλλου, όπως εξηγήθηκε προηγουμένως. Οι εικονικές μηχανές είναι ιδανικές για εκτέλεση εφαρμογών και προγραμμάτων με ευελιξία και απομόνωση περιβάλλοντος, ενώ οι εξυπηρετητές γυμνού μετάλλου παρέχουν πιο άμεσο έλεγχο στα βαθύτερα επίπεδα του υλικού. (βλ. Κ.2.1) .

Σε κάθε εικονική μηχανή εκτελείται ένα μοναδικό φιλοξενούμενο λειτουργικό σύστημα (Guest OS) που μπορεί να είναι διαφορετικό από αυτό (Host OS) στο οποίο εκτελείται ο υπερεπόπτης. Η δυνατότητα των εικονικών μηχανών να εκτελούν διαφορετικό φιλοξενούμενο λειτουργικό σύστημα και κατά επέκταση πυρήνα (kernel), συνιστά ένα πολύ σημαντικό πλεονέκτημα ευελιξίας, απομόνωσης, ασφάλειας κ.α., το οποίο δεν υποστηρίζεται στους περιέκτες. Από την άλλη μεριά, αυτή η δυνατότητα έχει σαν αποτέλεσμα να προστίθεται υπερπληροφορία (overhead) στους φυσικούς νεφούπολογιστικούς πόρους, αφού δεσμεύονται από τον υπερεπόπτη για το φιλοξενούμενο λειτουργικό σύστημα, είτε χρησιμοποιούνται, είτε όχι. Η στατική ανάθεση πόρων, στις εικονικές μηχανές, ενώ αρχικά φαίνεται να μη δημιουργεί κάποιο πρόβλημα, στην πραγματικότητα δεν ικανοποιεί την απαίτηση για βέλτιστη χρήση των νεφούπολογιστικών πόρων.

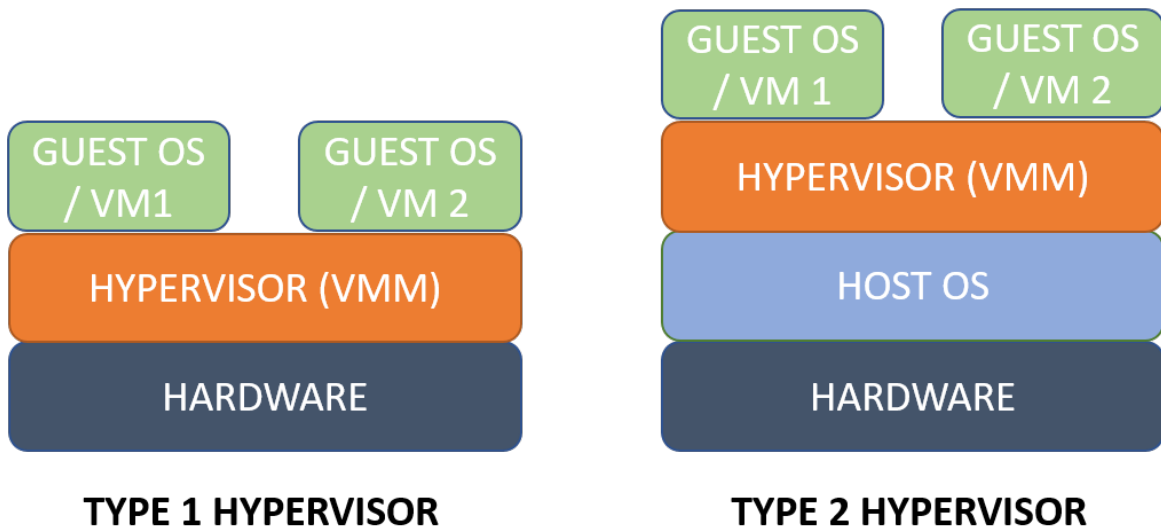
Για τη συντήρηση του λειτουργικού συστήματος και των εφαρμογών που εκτελούνται σε εικονικές μηχανές, απαιτούνται τακτικοί έλεγχοι και διαδικασίες διαχείρισης από τον οργανισμό. Σε περιπτώσεις όπου υπάρχουν συγκεκριμένες κατηγορίες υπηρεσιών, συνήθως είναι προτιμότερο να χρησιμοποιείται μια υπηρεσία Πλατφόρμας ως Υπηρεσία (PaaS). Για παράδειγμα, σε μια βάση δεδομένων, η διαχείριση και η συντήρηση μιας υποδομής μπορεί να απαιτεί περισσότερο χρόνο και προσπάθεια από ό,τι όταν παρέχεται αυτοματοποιημένα, από έναν πάροχο υπηρεσιών νέφους. Σε αυτήν την περίπτωση, ο οργανισμός απλά αξιοποιεί την παρεχόμενη υπηρεσία βάσης δεδομένων, ενώ οι διαδικασίες διαχείρισης και συντήρησης εκτελούνται από τον πάροχο υπηρεσιών νέφους. Αυτό επιτρέπει στον οργανισμό να επικεντρωθεί στη χρήση και την αξιοποίηση της βάσης δεδομένων, αποφεύγοντας τις διαδικασίες διαχείρισης της υποκείμενης υποδομής. [21][10]

2.3.1 Υπερεπτόπτες Εικονικών Μηχανών

Οι εικονικές μηχανές δημιουργούνται, είναι διαχειρίσιμες και γίνονται διαθέσιμες μέσω ενός υπερεπτόπτη (hypervisor). Ο υπερεπτόπτης είναι υπεύθυνος για την παρακολούθηση και τον έλεγχο των φυσικών πόρων που χρησιμοποιούνται από τις φιλοξενούμενες εικονικές μηχανές. Επιπλέον, ο υπερεπτόπτης είναι υπεύθυνος για την πολυπλοκότητα της ανάθεσης των υποκείμενων φυσικών πόρων στις εικονικές μηχανές που φιλοξενούνται. Οι υπερεπτόπτες εικονικών μηχανών χωρίζονται συνήθως σε δύο κύριες κατηγορίες

- Type 1 Hypervisor (Bare Metal)
- Type 2 Hypervisor (Hosted)

Οι Τύπου 1 Υπερεπτόπτες εγκαθίσταται απευθείας στο υλισμικό, ενώ οι Τύπου 2 Υπερεπτόπτες εγκαθίστανται στο λειτουργικό σύστημα που εκτελείται στο υλισμικό. Ο υπερεπτόπτης αναλαμβάνει το ρόλο της αφαίρεσης (abstraction) του υλισμικού και της προσφοράς του, μέσω εικονικοποιημένων πόρων, στις εικονικές μηχανές [22] [23].



Σχήμα 2. Υπερεπτόπτες εικονικών μηχανών

2.3.1.1 Υπερεπτόπτες Τύπου 1 (Type 1 Hypervisors)

Οι Τύπου 1 Υπερεπτόπτες είναι οι πιο συνήθεις υλοποιήσεις, όπου το λογισμικό εικονικοποίησης εγκαθίσταται απευθείας στο υλισμικό, και λειτουργεί ως ένα επίπεδο αφαίρεσης μεταξύ του υλικού και των εικονικών μηχανών, όπως συμβαίνει όταν εγκαθίστανται ένα οποιοδήποτε λειτουργικό σύστημα. Επειδή οι Τύπου 1 Υπερεπτόπτες είναι ανεξάρτητοι από οποιοδήποτε λειτουργικό σύστημα, είναι εξαιρετικά ασφαλείς, αφού δεν κινδυνεύουν από επιθέσεις σε πιθανές ευλάβειες που μπορεί να έχουν τα λειτουργικά συστήματα. Επιπροσθέτως, οι Τύπου 1 Υπερεπτόπτες, συνήθως, παρουσιάζουν καλύτερη απόδοση και αποτελεσματικότητα σε σύγκριση με τους Τύπου 2 Υπερεπτόπτες, καθώς έχουν πιο άμεση πρόσβαση και έλεγχο στους φυσικούς πόρους του συστήματος και κατά επέκταση λιγότερη υπερπληροφορία (overhead) [22].

Ενδεικτικά παραδείγματα Υπερεπτοπών Τύπου 1 είναι τα:

- VMware vSphere / ESXi
- Xen / Citrix XenServer
- Red Hat Enterprise Virtualization (RHEV) / oVirt
- Microsoft Hyper-V
- KVM (hybrid)*

* Το KVM (Kernel-based Virtual Machine) αναφέρεται ως «υβριδικό» (hybrid), καθώς συνδυάζει τα χαρακτηριστικά από τους Τύπου 1 και Τύπου 2 υπερεπόπτες. Αυτό επιτυγχάνεται μέσω του εργαλείου (module) KVM που ενσωματώνεται στον πυρήνα του Linux, μετατρέποντας τον σε υπερεπόπτη Τύπου 1. Αυτό σημαίνει ότι το KVM μπορεί να εκμεταλλεύεται απευθείας τις δυνατότητες του υλικού και να προσφέρει υψηλή απόδοση και αξιοπιστία. [24]

Παράλληλα, από την οπτική γωνία του λειτουργικού συστήματος, το KVM μπορεί να θεωρηθεί ως υπερεπόπτης Τύπου 2, καθώς το Linux λειτουργικό σύστημα υποδοχής λειτουργεί πλήρως και οι εικονικές μηχανές αντιμετωπίζονται ως κανονικές διεργασίες του συστήματος. Αυτό σημαίνει ότι οι εικονικές μηχανές μπορούν να χρησιμοποιήσουν τους πόρους του λειτουργικού συστήματος για την εκτέλεσή τους.

Παρόλα αυτά, γενικά το KVM ταξινομείται στους υπερεπόπτες τύπου 1, καθώς επιτρέπει την εκτέλεση εικονικών μηχανών απευθείας στο υλικό, παρέχοντας τους απευθείας πρόσβαση σε φυσικούς πόρους, όπως επεξεργαστές και μνήμη [25]. Αυτή η αρχιτεκτονική του επιτρέπει να παρέχει απόδοση και αξιοπιστία παρόμοια με αυτήν που εμφανίζουν οι Τύπου 1 υπερεπόπτες.

2.3.1.2 Υπερεπόπτες Τύπου 2 (Type 2 Hypervisors)

Οι Τύπου 2 Υπερεπόπτες εκτελούνται στο ήδη εγκατεστημένο λειτουργικό σύστημα της μηχανής υποδοχής (Host Machine), ενώ οι Τύπου 1 Υπερεπόπτες εγκαθίστανται απευθείας στο υλισμικό. Παρά το γεγονός ότι εκτελούνται στο ήδη εγκατεστημένο λειτουργικό σύστημα, μπορούν να υποστηρίξουν διαφορετικά λειτουργικά συστήματα από αυτά της μηχανής υποδοχής. Ένα αρνητικό των Τύπου 2 υπερεπόπτων είναι η καθυστέρηση (latency) που προκύπτει, αφού ο υπερεπόπτης δεν επικοινωνεί απευθείας με το υλισμικό, αλλά με το λειτουργικό σύστημα. Το λειτουργικό σύστημα μεταφράζει την όποια διαδικασία, άρα αυτόματα δημιουργείται ένα επιπλέον επίπεδο καθυστέρησης [22].

Ενδεικτικά παραδείγματα Υπερεποπτων Τύπου 2 είναι τα

- VirtualBox
- VMware Workstation

- Microsoft Virtual PC

2.4 Περιέκτες

Η τεχνολογία Περιέκτη έχει ωριμάσει αρκετά και αποτελεί πλέον ένα κυρίαρχο εργαλείο στον τομέα της τεχνολογίας. Ορίζεται ως εικονικοποίηση βασισμένη στο λειτουργικό σύστημα [26] και μπορεί να περιγραφεί ως μία μονάδα πακετοποίησης (packaging unit) της ίδιας της εφαρμογής και των εξαρτήσεων της. Με τους περιέκτες δημιουργούνται οι κατάλληλες συνθήκες για εύκολη και ευέλικτη διαχείριση του κύκλου ζωής οποιασδήποτε εφαρμογής.

Κολοσσοί στον τομέα της πληροφορικής, όπως η Google, η Amazon, η Microsoft και άλλοι, έχουν ενσωματώσει την τεχνολογία των περιεκτών στα κέντρα δεδομένων τους και μέσω αυτής της τεχνολογίας, έχουν καταφέρει να διαχειρίζονται τις νεφούπολογιστικές υποδομές τους με πιο εύκολο, γρήγορο και αποτελεσματικό τρόπο. Οι περιέκτες υποστηρίζουν την απομόνωση των εφαρμογών και των περιβαλλόντων εκτέλεσής τους, επιτρέποντας την ευκολότερη ανάπτυξη, τη μεταφορά και την εκτέλεση εφαρμογών σε διάφορες πλατφόρμες.

Οι περιέκτες λειτουργούν, ενδογενώς, ως διεργασίες (processes) στο λειτουργικό σύστημα υποδοχής (Host OS), κάνοντας χρήση του πυρήνα του. Αυτό σημαίνει ότι κάθε περιέκτης δεν εκτελεί ένα μοναδικό ανεξάρτητο λειτουργικό σύστημα, αλλά λειτουργεί ως μια διεργασία εντός του λειτουργικού συστήματος υποδοχής. Αυτή η δυνατότητα εξαλείφει την ανάγκη για στατική ανάθεση πόρων και περιορίζει το πρόβλημα της υπερπληροφορίας. Με αυτόν τον τρόπο, δημιουργούνται οι κατάλληλες συνθήκες για την αρτιότερη αξιοποίηση των νεφούπολογιστικών πόρων, προσφέροντας οφέλη τόσο στους χρήστες (tenants) όσο και στους παρόχους υπηρεσιών νέφους. Αυτό συμβαδίζει με τους 10 νόμους των cloudonomics, που προάγουν την οικονομική αποδοτικότητα και τη βέλτιστη αξιοποίηση των πόρων στον τομέα του νέφους.

Ο περιέκτης αποτελεί μια απομονωμένη (isolated) μονάδα, στην οποία το λογισμικό και οι εξαρτήσεις του έχουν πακετοποιηθεί, επιτρέποντας την αξιόπιστη λειτουργία του σε οποιοδήποτε περιβάλλον. Οι περιέκτες παρέχουν μεγάλη φορητότητα, αντίθετα από τις παραδοσιακές μεθόδους ανάπτυξης λογισμικού, όπου η μεταφορά τους σε διαφορετικό περιβάλλον απαιτεί τροποποιήσεις και μπορεί να προκαλέσει σφάλματα ή ασυμβατότητες. Ο περιέκτης περιλαμβάνει όλα

τα απαραίτητα στοιχεία που απαιτούνται για την εκτέλεση του λογισμικού, όπως βιβλιοθήκες, πακέτα και ρυθμίσεις, επιτρέποντας στο λογισμικό να λειτουργεί απομονωμένα και αξιόπιστα, ανεξάρτητα από τις υπόλοιπες εφαρμογές ή τις διαθέσιμες πλατφόρμες. Έτσι, ο περιέκτης επιτρέπει την εύκολη μεταφορά και εκτέλεση του λογισμικού σε διαφορετικά συστήματα και περιβάλλοντα χωρίς προβλήματα συμβατότητας ή ανεπιθύμητες επιπτώσεις.

Η φορητότητα, που προσφέρουν οι περιέκτες, επιτρέπει στις ομάδες ανάπτυξης να απολαμβάνουν μεγαλύτερη ευελιξία και ταχύτητα στον κύκλο ζωής του λογισμικού. Καθώς οι περιέκτες εξασφαλίζουν ότι το λογισμικό θα εκτελείται σωστά, απαλλαγμένο από σφάλματα και αστοχίες, σε διάφορες πλατφόρμες, μειώνουν την ανάγκη για πολύπλοκες διαδικασίες ανάπτυξης και δοκιμών. Επιπλέον, η απομόνωση των περιεκτών προσφέρει ασφάλεια και προστασία από πιθανές απειλές και επιθέσεις, καθώς τυχόν προβλήματα που προκαλούνται από το ένα λογισμικό δεν επηρεάζουν τα υπόλοιπα περιβάλλοντα.

Δύο βασικές τεχνολογίες που χρησιμοποιούνται στον κόσμο των περιεκτών, για την απομόνωση και την διαχείριση των υποκείμενων πόρων, είναι τα `cgroups` (`control groups`) και τα `namespaces`.

Τα `cgroups` είναι ένα μηχανισμός που επιτρέπει τον περιορισμό και τη διαχείριση των διαθέσιμων πόρων, όπως η ΚΜΕ, η μνήμη, ο χώρος αποθήκευσης και το δίκτυο. Με τη χρήση των `cgroups`, μπορούν να οριστούν περιορισμοί και προτεραιότητες στους περιέκτες, εξασφαλίζοντας ότι οι πόροι κατανέμονται αποτελεσματικά. Αυτό επιτρέπει τη βέλτιστη αξιοποίηση των πόρων του συστήματος και την αποφυγή της υπερβολικής κατανάλωσης από έναν περιέκτη.

Τα `namespaces`, από την άλλη πλευρά, παρέχουν μια μορφή απομόνωσης του περιβάλλοντος ενός περιέκτη. Επιτρέπουν σε μια διεργασία να έχει μια διαφορετική άποψη του συστήματος από ό,τι άλλες διεργασίες, περιορίζοντας την ορατότητα και την πρόσβασή της σε πόρους και λειτουργίες του συστήματος. Κάθε περιοχή περιορισμού, δηλαδή κάθε περιέκτης, μπορεί να έχει τα δικά της `namespaces` για τα δίκτυα, τις διεργασίες, τον χρόνο εκτέλεσης και άλλες πτυχές του περιβάλλοντος, προσφέροντας έτσι απομόνωση και ασφάλεια μεταξύ των περιεχομένων περιεκτών.

Λόγω της μεγάλης ανάπτυξης, στο πεδίο των περιεκτών, χρειάστηκε να οριστούν κάποια πρότυπα (standards), ώστε να υπάρξει ευρύτερη διαλειτουργικότητα στις πλατφόρμες περιεκτών, ανεξάρτητα από το ποια μηχανή ή ποιο περιβάλλον εκτέλεσης θα επιλεγεί κάθε φορά. Έτσι, το 2015 δημιουργήθηκε το OCI (Open Container Initiative), το οποίο ασχολήθηκε με τρεις κύριους τομείς του οικοσυστήματος των περιεκτών. Προσδιόρισε την δομή των μηχανών περιέκτη και των περιβαλλόντων εκτέλεσης (runtime) τους, τη μορφή (format) των εικόνων περιέκτη και τον τρόπο διανομής των εικόνων.

Οι περιέκτες αν και μοιάζουν αρκετά με τις εικονικές μηχανές, είναι αρκετά πιο ελαφρείς, απαιτούν λιγότερους υπολογιστικούς πόρους και είναι κατά πολύ ταχύτεροι στην εκκίνηση τους. Τεχνικά, οι περιέκτες διαφέρουν από τις εικονικές μηχανές καθώς μοιράζονται τον πυρήνα του λειτουργικού συστήματος με άλλους περιέκτες και εφαρμογές, ενώ οι εικονικές μηχανές λειτουργούν δημιουργώντας ένα ξεχωριστό και πλήρες εικονικό λειτουργικό σύστημα. Αυτή η διαφορά, στον τρόπο λειτουργίας, επιτρέπει στους περιέκτες να είναι πιο ελαφριοί και πιο αποδοτικοί στη χρήση των πόρων, καθιστώντας τους ιδανικούς για εφαρμογές που απαιτούν γρήγορη εκκίνηση και χαμηλότερη κατανάλωση ενέργειας. [21] [27]

2.4.1 Μηχανές και Περιβάλλοντα Εκτέλεσης Περιεκτών

Η τεχνολογία περιέκτη έχει γνωρίσει τεράστια ανάπτυξη και διάδοση, προσφέροντας μια πληθώρα λύσεων για την επιλογή της κατάλληλης τεχνολογίας περιέκτη. Το Docker, χωρίς αμφιβολία, έχει κατακτήσει τη θέση του πιο γνωστού και ευρέως χρησιμοποιούμενου εργαλείου στον τομέα αυτό. Ωστόσο, υπάρχουν και άλλες αξιόλογες τεχνολογίες περιεκτών, κάθε μία απευθυνόμενη σε συγκεκριμένες χρήσεις και προσφέροντας διαφορετικά πλεονεκτήματα [28].

Για παράδειγμα, το LXD παρέχει υψηλή απόδοση και ασφάλεια, επιτρέποντας την εκτέλεση πολλαπλών περιεκτών σε φυσικές ή εικονικές μηχανές. Το CRI-O είναι μια ελαφριά υλοποίηση, που εστιάζει στην υποστήριξη του προτύπου Kubernetes Container Runtime Interface (CRI). Το runC είναι η βασική τεχνολογία πίσω από το Docker και παρέχει ένα περιβάλλον εκτέλεσης περιεκτών, ενώ το containerd είναι ένας αυτόνομος δαίμονας για τη διαχείριση περιεκτών.

Άλλες τεχνολογίες περιεκτών περιλαμβάνουν το rkt, μια μηχανή περιέκτη η οποία επικεντρώνεται στην ασφάλεια και την απλότητα, καθώς και τα Kata containers,

Nabla containers, Podman και Singularity (Apptainer), κάθε ένα από τα οποία έχει τις ιδιαιτερότητές του και στοχεύει σε συγκεκριμένες περιπτώσεις χρήσης.

Οι περιέκτες για την λειτουργία τους απαιτούν ένα περιβάλλον εκτέλεσης υψηλού επιπέδου (μηχανή περιέκτη). Οι μηχανές περιεκτών παραδίδουν τον περιέκτη σε ένα χαμηλού επιπέδου περιβάλλον εκτέλεσης περιεκτών. Βάσει αυτού, τα περιβάλλοντα εκτέλεσης περιεκτών χωρίζονται, συχνά, σε δύο κατηγορίες, τα υψηλού επιπέδου (μηχανή περιέκτη) και τα χαμηλού επιπέδου. Η διαφορά έγκειται στο βάθος υποστήριξης των προδιαγραφών του OCI (Open Container Initiative) και στα πρόσθετα χαρακτηριστικά που περιλαμβάνουν [29]. Τα περιβάλλοντα εκτέλεσης περιεκτών είναι υπεύθυνα για τη διαχείριση τόσο των εικόνων περιέκτη, όσο και των ίδιων των περιεκτών, για τη δημιουργία (build) εικόνων, για το διαμοιρασμό των εικόνων (π.χ. μέσω κάποιου αποθετηρίου) αλλά και για την εκτέλεση των περιεκτών. Τα χαμηλού επιπέδου περιβάλλοντα εκτέλεσης περιεκτών (runC, containerd κ.α.) υλοποιούν την ίδια την εκτέλεση των περιεκτών και διαχειρίζονται τα cgroups και τα namespaces, χωρίς να εμπλέκονται με τις υπόλοιπες λειτουργίες που αναφέρθηκαν. Με τις υπόλοιπες διαδικασίες ασχολούνται οι μηχανές περιέκτη (rkt, Podman, CRI-O κ.α.). [30]

Παράλληλα με το Docker, αναδύθηκε μια εξίσου σημαντική τεχνολογία περιεκτών, η οποία έχει αποκτήσει μεγάλη απήχηση στον τομέα της ενορχήστρωσης των περιεκτών. Αυτή η τεχνολογία, γνωστή ως Kubernetes, αναπτύχθηκε αρχικά από την Google. Ένα από τα βασικά χαρακτηριστικά του Kubernetes είναι ότι δεν περιλαμβάνει ένα προκαθορισμένο περιβάλλον εκτέλεσης, αλλά επιτρέπει την επιλογή του περιβάλλοντος που επιθυμεί ο χρήστης για την εκάστοτε περίπτωση. Αυτό το χαρακτηριστικό του Kubernetes οδήγησε το 2016 στην εισαγωγή μιας πρόσθετης διεπαφής προγραμματισμού εφαρμογών (API plugin), γνωστής ως CRI, που επιτρέπει την υποστήριξη πολλαπλών περιβαλλόντων εκτέλεσης περιέκτη, χωρίς να χρειάζεται το Kubernetes να γνωρίζει πως αυτά λειτουργούν.

Έτσι, το Kubernetes μπορεί να αλληλοεπιδρά με διάφορα περιβάλλοντα εκτέλεσης περιεκτών όπως το Docker, το containerd, το rkt και άλλα, παρέχοντας μεγαλύτερη ευελιξία και επιλογές στους χρήστες. Αυτό επιτρέπει την επικέντρωση στην ενορχήστρωση και στην διαχείριση των περιεκτών, ενώ ταυτόχρονα επιτρέπει

την αξιοποίηση των δυνατοτήτων και των πλεονεκτημάτων κάθε περιβάλλοντος εκτέλεσης.

Στη συνέχεια, θα αναλυθούν, το Docker και το LXC, δύο θεμέλιες τεχνολογίες που έχουν συμβάλει σημαντικά στην εξέλιξη των περιεκτών, καθώς και τρεις υλοποιήσεις του CRI το containerd, το CRI-O και το runC, που υποστηρίζονται από το Kubernetes και το CNCF (Cloud Native Computing Foundation), οργανισμό στον οποίο φιλοξενούνται διάφορα νεφούπολογιστικά έργα.

Η πρώτη τεχνολογία είναι το LXC (LinuX Containers), το οποίο εμφανίστηκε για πρώτη φορά το 2008. Το LXC είναι ένα σύστημα που επιτρέπει τη δημιουργία και διαχείριση περιεκτών σε επίπεδο λειτουργικού συστήματος Linux. Η δεύτερη τεχνολογία περιέκτη που θα αναλυθεί είναι το Docker που είναι και η πιο διαδεδομένη. Το Docker είναι μια πλατφόρμα περιεκτών που μπορεί να θεωρηθεί, κατά μία έννοια, διάδοχος του LXC. Η κύρια διαφορά του Docker είναι ότι παρέχει ένα επιπλέον επίπεδο αφάιρησης (abstraction layer), το Docker Engine, σε σχέση με τους LXC περιέκτες, καθιστώντας τη διαδικασία δημιουργίας και διαχείρισης περιεκτών πιο ευέλικτη και ευκολότερη. Ακολούθως θα αναλυθεί το containerd, το οποίο είναι μια ανεξάρτητη υλοποίηση του περιβάλλοντος εκτέλεσης Docker και παρέχει λύσεις για τη δημιουργία και τη διαχείριση περιεκτών. Έπειτα το CRI-O, το οποίο είναι μια απλοποιημένη και ελαφριά υλοποίηση του CRI, σχεδιασμένη αποκλειστικά για το Kubernetes. Τέλος, το runC που είναι μια πρότυπη υλοποίηση του OCI περιβάλλοντος εκτέλεσης για τη δημιουργία και τη διαχείριση περιεκτών. [31]

2.4.1.1 LXC (LinuX Containers)

Το LXC αποτελεί την εξέλιξη του OpenVZ και βασίζεται στις αρχές και τη φιλοσοφία του αρχικού έργου (project). Η κύρια λειτουργία του LXC είναι να προσφέρει εικονικοποίηση σε επίπεδο λειτουργικού συστήματος, επιτρέποντας τη δημιουργία πολλαπλών απομονωμένων εικονικών περιβαλλόντων Linux (Linux Virtual Environment) σε ένα μόνο μηχάνημα, είτε αυτό είναι φυσικό, είτε εικονικό. Τα εικονικά περιβάλλοντα αποτελούν περιέκτες που μπορούν, είτε να πακετοποιούν και να διαχειρίζονται εφαρμογές, διασφαλίζοντας την αυτονομία τους και την απομόνωσή τους από το υπόλοιπο σύστημα, είτε να προσομοιώνουν καινούργιες, πλήρεις εικονικές μηχανές με το δικό τους λειτουργικό σύστημα [32].

Είναι ουσιώδες να τονιστεί η διαφορά μεταξύ των εικονικών περιβαλλόντων και των εικονικών μηχανών. Η κύρια διαφορά αυτών των δύο τεχνολογιών αφορά την απουσία υπερεπόπτη (hypervisor) στα εικονικά περιβάλλοντα, μιας συνιστώσας που αποτελεί κεντρικό στοιχείο της εξομοίωσης στις εικονικές μηχανές. Ειδικότερα, στα εικονικά περιβάλλοντα το μόνο που εξομοιώνεται είναι το λειτουργικό σύστημα που εκτελείται μέσα στον περιέκτη, ενώ δεν υπάρχει καμία εξομοίωση υλισμικού. Αυτή η διαφορά αποτελεί κρίσιμη παράμετρο, καθώς επιτρέπει την εξαιρετικά γρήγορη δημιουργία ενός περιέκτη, σε αντίθεση με τις εικονικές μηχανές που απαιτούν πολύ περισσότερο χρόνο λόγω της εξομοίωσης του υλισμικού.

Επιπρόσθετα, στα εικονικά περιβάλλοντα η εκκίνηση των εφαρμογών γίνεται άμεσα, μειώνοντας τον χρόνο που απαιτείται για την εκτέλεση και την ανάπτυξη των λειτουργιών, συνεισφέροντας έτσι στην αποτελεσματικότητα και την παραγωγικότητα της διαδικασίας.

2.4.1.1.1 LXC εργαλεία (tools used with LXC)

Χρησιμοποιώντας την τεχνολογία SSH (Secure Shell), προσφέρεται η δυνατότητα πρόσβασης στο LXC, δηλαδή με τον ίδιο τρόπο που πραγματοποιούνται οι συνδέσεις σε έναν εξυπηρετητή γυμνού μετάλλου ή σε μια εικονική μηχανή. Αυτή η ομοιομορφία στην πρόσβαση και στην διαχείριση διευκολύνει σημαντικά την μεταφορά εφαρμογών στα εικονικά περιβάλλοντα.

Για να δημιουργηθεί ένας περιέκτης, χρειάζεται μια LXC εικόνα (image). Το LXC μπορεί να θεωρηθεί ως ένας υπερεπόπτης για περιέκτες, ο οποίος με τη βοήθεια κάποιου API, διευρύνει τη λειτουργικότητα του LXC, προσφέροντας δυνατότητες όπως μετακόμιση ή δημοσίευση εικόνων (publishing images) περιεκτών [33]. Όταν βάσει μιας εικόνας, δημιουργηθεί ένας περιέκτης και εισέλθει σε λειτουργία, τότε οι διάφορες διαχειριστικές λειτουργίες μπορούν να γίνουν μέσω της διεπαφής γραμμής εντολών (CLI). Η ευελιξία και το βάθος παραμετροποίησης που προσφέρει το LXC CLI είναι καίριας σημασίας για την αποτελεσματική διαχείριση των περιεκτών, σε ένα εικονικοποιημένο περιβάλλον. Η διεπαφή γραμμής εντολών του LXC παρέχει ένα ευρύ φάσμα βασικών εντολών, που επιτρέπουν στον διαχειριστή να εκτελεί ολοκληρωμένα τις πιο σημαντικές λειτουργίες στους περιέκτες. Μερικές από αυτές τις λειτουργίες είναι η δημιουργία, η εκτέλεση, η παύση λειτουργίας, η

συνέχιση, η διαγραφή περιεκτών και άλλες, επιβεβαιώνοντας τη δυνατότητα για πλήρη και ευέλικτη διαχείριση των LXC περιεκτών.

Οι περιέκτες LXC λειτουργούν ως ανεξάρτητα στιγμιότυπα ενός Linux λειτουργικού συστήματος. Αυτή η ιδιότητα, σημαίνει ότι η πλειοψηφία των εργαλείων και των εφαρμογών που χρησιμοποιούνται σε τυπικά Linux περιβάλλοντα είναι άμεσα συμβατά με τους LXC περιέκτες, αποφεύγοντας την ανάγκη για εξειδικευμένα εργαλεία τρίτων.

Για την διαχείριση περιεκτών που εκτελούνται σε διαφορετικούς εξυπηρετητές, το LXD, ένα ειδικό περιβάλλον εκτέλεσης (runtime) για LXC περιέκτες, προσφέρει APIs για δικτυακή διαχείριση. Το LXD προσφέρει ευελιξία και επεκτασιμότητα, επιτρέποντας την εύκολη ενσωμάτωση με μεγαλύτερες πλατφόρμες όπως το OpenStack και το Proxmox, ενώ παράλληλα υποστηρίζει τις αυξανόμενες ανάγκες των υποδομών νέφους.

Επίσης, το LXC προσφέρει πολύ καλή τεκμηρίωση, η οποία μπορεί να φανεί χρήσιμη, τόσο σε αρχάριους όσο και σε προχωρημένους χρήστες, κατά τις διαδικασίες δημιουργίας, διανομής και διαχείρισης περιεκτών. Το LXC προσφέρει υποστήριξη για πολλές γλώσσες προγραμματισμού, μέσω των διαθέσιμων βιβλιοθηκών του, όπως η Python και η Java που αποτελούν δύο από τα πιο κοινά παραδείγματα, χωρίς η υποστήριξη γλωσσών προγραμματισμού να εξαντλείται σε αυτές. Η υποστήριξη γλωσσών προγραμματισμού, διευρύνει την προσβασιμότητα του LXC σε διάφορες ομάδες ανάπτυξης λογισμικού, καθιστώντας την εφαρμογή και την διαχείριση των περιεκτών ευκολότερη και πιο ευέλικτη [32].

2.4.1.1.2 Πλεονεκτήματα του LXC

Το LXC προσφέρει, μεταξύ άλλων, ένα σημαντικό πλεονέκτημα σε σχέση με τις εικονικές μηχανές, που εντοπίζεται στην απουσία εξομοιωμένου υλικού, εικονικοποιημένου επεξεργαστή και εικονικοποιημένων διεπαφών. Στα LXC περιβάλλοντα, το μόνο στοιχείο που εξομοιώνεται είναι το ίδιο το λειτουργικό σύστημα. Αυτό οδηγεί σε αυξημένη απόδοση, καθώς οι περιέκτες λειτουργούν πολύ ταχύτερα σε σχέση με τις τυπικές εικονικές μηχανές.

Επιπρόσθετα, οι LXC περιέκτες χαρακτηρίζονται από αποδοτικότερη διαχείριση των πόρων της φιλοξενούμενης μηχανής. Η κατανάλωση επεξεργαστικής ισχύος (CPU) και μνήμης RAM είναι χαμηλότερη στους περιέκτες σε σύγκριση με τις

εικονικές μηχανές, κάνοντας την εφαρμογή τους ιδιαίτερα ελκυστική για σενάρια με περιορισμένους πόρους ή υψηλές απαιτήσεις απόδοσης.

Ένα από τα πλεονεκτήματα της τεχνολογίας LXC, έναντι της τεχνολογίας Docker, εντοπίζεται στην απλούστερη φιλοσοφία που ακολουθεί για την δημιουργία και την διαχείριση των περιεκτών. Οι περιέκτες LXC παρουσιάζουν μεγάλη ομοιότητα με τις εικονικές μηχανές, αφού διαθέτουν το δικό τους λειτουργικό σύστημα στο οποίο εκτελούνται τα πακέτα λογισμικού. Αυτό καθιστά τη μεταφορά μιας εφαρμογής από μια εικονική μηχανή ή έναν φυσικό εξυπηρετητή σε έναν LXC περιέκτη λιγότερο απαιτητική σε σχέση με τη μετάβαση σε έναν Docker περιέκτη. Στο περιβάλλον Docker, μια εφαρμογή ενδέχεται να χρειάζεται να αποδομηθεί σε μικρότερα κομμάτια, κάθε ένα από τα οποία θα πακετοποιηθεί σε έναν διαφορετικό περιέκτη.

Επιπλέον, η διαχείριση των LXC περιεκτών είναι γενικά λιγότερο απαιτητική, καθώς οι συστοιχίες από LXC περιέκτες είναι συνήθως λιγότερο πολύπλοκες στην υλοποίησή τους σε σχέση με τις αντίστοιχες συστοιχίες Docker. Αυτό σημαίνει, ότι η υποστήριξη και η συντήρηση των συστημάτων που βασίζονται σε LXC περιέκτες μπορεί να αποδειχθεί λιγότερο απαιτητική στα περισσότερα περιβάλλοντα.

2.4.1.1.3 Μειονεκτήματα του LXC

Στα μειονεκτήματα των LXC περιεκτών μπορούν να συμπεριληφθούν η ανάγκη για χειροκίνητη εγκατάσταση λογισμικού, μια διαδικασία που μοιάζει περισσότερο με την εγκατάσταση σε εικονικές μηχανές ή σε φυσικούς εξυπηρετητές. Επιπλέον, η διαχείριση και η παρακολούθηση των ενημερώσεων λογισμικού απαιτεί συνέπεια και επιμέλεια, μια διαδικασία που προϋποθέτει επιπλέον χρόνο και προσπάθεια.

Σε αντίθεση με το Docker, όπου η ανανέωση των εφαρμογών μπορεί να γίνει με τη δημιουργία νέων περιεκτών που περιλαμβάνουν τις ενημερωμένες εκδόσεις των εφαρμογών [32], στην LXC τεχνολογία, η ενημέρωση λογισμικού είναι μια πιο περίπλοκη διαδικασία. Προκειμένου να εγκαταστήσουν ή να αναβαθμίσουν μια εφαρμογή, οι διαχειριστές πρέπει να εισέλθουν σε κάθε περιέκτη ξεχωριστά, με τον ίδιο τρόπο όπως θα έκαναν σε έναν φυσικό εξυπηρετητή ή σε μια εικονική μηχανή.

Ένα, ακόμη, σημαντικό μειονέκτημα των LXC περιεκτών είναι η περιορισμένη δυνατότητα υποστήριξης διαφορετικών λειτουργικών συστημάτων πέρα από το Linux. Αυτό είναι αποτέλεσμα του γεγονότος ότι η τεχνολογία LXC βασίζεται στην εξομίωση που παρέχεται από τις δυνατότητες του πυρήνα του Linux, όπως τα

namespaces και τα cgroups [34]. Συνεπώς, τα λειτουργικά συστήματα που δεν βασίζονται στον πυρήνα του Linux δεν μπορούν να εκτελεστούν εντός ενός LXC περιέκτη. Αυτό θα μπορούσε να προκαλέσει προβλήματα εάν, για παράδειγμα, εμφανιζόταν η ανάγκη να εκτελεστεί ένα λειτουργικό σύστημα Windows ή macOS εντός ενός περιέκτη, δεν θα ήταν εφικτό με την τεχνολογία LXC. Επίσης, οι παραλλαγές του πυρήνα Linux που δεν συμμορφώνονται πλήρως με τις προδιαγραφές του Linux, μπορεί να αντιμετωπίσουν δυσκολίες κατά την εξομίωση σε έναν LXC περιέκτη, προκαλώντας περαιτέρω προβλήματα συμβατότητας και απόδοσης. Καθώς η τεχνολογία LXC χρησιμοποιεί τα namespaces για τον αμοιβαίο αποκλεισμό των περιεκτών από το λειτουργικό σύστημα που τους φιλοξενεί, το επίπεδο απομόνωσης και ασφάλειας που παρέχεται δεν είναι τόσο ισχυρό όσο εκείνο που παρέχεται από τις εικονικές μηχανές. Αυτό συνιστά μια σημαντική ανησυχία για την ασφάλεια, καθώς η ευκολία με την οποία οι επιθέσεις μπορούν να διαδοθούν μεταξύ των LXC περιεκτών φαντάζει αρκετά μεγάλη.

Εάν το λειτουργικό σύστημα που φιλοξενεί τους περιέκτες είναι ευάλωτο σε επιθέσεις, οι επιτιθέμενοι μπορούν να εκμεταλλευτούν αυτήν την ευπάθεια για να επεκτείνουν την επίθεση τους σε άλλους περιέκτες στο ίδιο σύστημα. Αυτό, είναι αρκετά επίφοβο ειδικά σε περιβάλλοντα πολλαπλών χρηστών, όπου οι επιτιθέμενοι θα μπορούσαν να προσπαθήσουν να επεκτείνουν την επίθεση τους σε περιέκτες που ανήκουν σε άλλους χρήστες ή εφαρμογές. Αν και τα namespaces, όπως και άλλα μέσα ασφάλειας των Linux, μπορούν να μετριάσουν αυτό το κίνδυνο, δεν μπορούν να τον εξαλείψουν εντελώς.

2.4.1.2 Docker

Το Docker ήταν αρχικά γνωστό ως dotCloud [32] και αυτή η τεχνολογία ήταν ένα παράπλευρο έργο (side project). Αυτό άλλαξε το 2013 όταν το project έγινε ανοιχτού κώδικα, φτάνοντας έτσι στο ευρύ κοινό. Το Docker βασίζεται στην γλώσσα προγραμματισμού GO. Η βασική ιδέα του Docker προέρχεται από την τεχνολογία του LXC, αλλά σταδιακά έχει επεκτείνει τις ιδιότητες αυτής της τεχνολογίας και την έχει εμπλουτίσει με νέες δυνατότητες, όσον αφορά την πακετοποίηση. Μία από τις βασικές προσθήκες, που έχει επιτύχει, είναι η χρήση API υψηλού επιπέδου, η οποία προσφέρει μια πιο αποδοτική λύση εικονικοποίησης. Με την προσθήκη αυτή δεν εκτελούνται σε απομόνωση ολόκληρα λειτουργικά συστήματα αλλά διακριτές

διεργασίες. Συμπερασματικά, οι Docker περιέκτες δεν υλοποιούν το δικό τους λειτουργικό σύστημα, αλλά βασίζονται στο λειτουργικό σύστημα υποδοχής και στις ιδιότητες του, και στη γενικότερη υλοποίηση της υποδομής που τα φιλοξενεί.

Το Docker μπορεί να χαρακτηριστεί ως μια πλατφόρμα περιέκτη με πολύ καλή φορητότητα (portability) της οποίας ο κύριος ρόλος είναι να πακετοποιεί την εφαρμογή και όλες τις εξαρτήσεις (dependencies) της σε έναν εικονικό περιέκτη. Ένας τέτοιος περιέκτης μπορεί με τη σειρά του να εκτελεστεί σε οποιοδήποτε Linux εξυπηρετητή. Το Docker μοιράζεται με το LXC τα cgroups, τα namespaces και το Linux πυρήνα. Συνολικά όμως οι Docker περιέκτες είναι αρκετά πιο εξελιγμένοι σε σχέση με τους LXC. Η τεχνολογία Docker επιτρέπει την δημιουργία ενός αντικειμένου, το οποίο μπορεί να περιέχει πολλές εφαρμογές, οι οποίες πακετοποιούνται μαζί. Αυτό το αντικείμενο μπορεί ακολούθως να εγκατασταθεί σε οποιοδήποτε μηχάνημα υποστηρίζει την Docker τεχνολογία. Το Docker περιέχει versioning σύστημα, παρόμοιο με του Git, χάρη στο οποίο είναι δυνατό να αποθηκεύσει διακριτές εκδόσεις του περιέκτη, να αναγνωρίσει συγκεκριμένες αλλαγές μεταξύ εκδόσεων, να σώσει αλλαγές που έχουν γίνει, να αναστρέψει αλλαγές επιστρέφοντας σε προηγούμενη έκδοση κ.α..

Μια επίσης πολύ χρήσιμη λειτουργία, που υποστηρίζεται από το Docker, είναι ότι δεν χρειάζεται πάντα να δημιουργούνται καινούργια αντικείμενα και πακέτα, όταν χρειάζεται να δημιουργηθούν αντίγραφα κάποιων υπηρεσιών. Για παράδειγμα, εάν απαιτείται να υπάρχουν περισσότεροι περιέκτες, με βάσεις δεδομένων MySQL ή δικτυακούς εξυπηρετητές Apache, δεν χρειάζεται κάθε φορά να δημιουργείται ένας καινούργιος περιέκτης και να πακετοποιούνται οι υπηρεσίες σε αυτόν. Αντιθέτως, μπορεί να δημιουργηθεί ένας περιέκτης βάσης (base container), κάτι σαν τα template στις εικονικές μηχανές, ο οποίος περιέχει την επιθυμητή υπηρεσία, και ακολούθως να αναπυχθεί ο επιθυμητός αριθμός αντιγράφων από τον περιέκτη βάσης, ο οποίος περιέχει ήδη την πακετοποιημένη υπηρεσία. Ένα επιπλέον πολύ ισχυρό εργαλείο, που ενισχύει την υιοθέτηση της Docker τεχνολογίας, είναι η ύπαρξη μίας δημόσιας βάσης δεδομένων (αποθετήριο) η οποία αποκαλείται Docker Hub και περιέχει ένα τεράστιο αριθμό από ήδη υλοποιημένους περιέκτες και εικόνες περιέκτη, που έχουν προσφερθεί από διάφορους χρήστες ανά την υφήλιο.

Το κύριο χαρακτηριστικό του Docker είναι ότι εγκατέλειψε τη παραδοσιακή εικονικοποίηση και την απομόνωση ολόκληρου του λειτουργικού συστήματος,

χαρακτηριστικό της LXC τεχνολογίας, και ότι εστίασε, κυρίως, στις ίδιες τις εφαρμογές και τις υπηρεσίες και στην απομόνωση αυτών [32], μέσω του Docker περιβάλλοντος που ανέπτυξε σταδιακά, γνωστό και ως libcontainer. Το Docker προσφέρει ένα περιβάλλον το οποίο αποτελείται από το λειτουργικό σύστημα υποδοχής, πάνω στο οποίο όλες οι εφαρμογές εκτελούνται σε πακετοποιημένους περιέκτες, με την κάθε εφαρμογή να έχει το δικό της απομονωμένο περιβάλλον. Συνήθως αυτό το περιβάλλον ορίζεται και δημιουργείται χρησιμοποιώντας μια εικόνα Docker. Η δημιουργία αυτού του περιβάλλοντος μπορεί να αυτοματοποιηθεί με τη βοήθεια αυτού που ονομάζουμε Dockerfile. Η Docker μηχανή (engine) χρησιμοποιείται για τη διαχείριση και τον έλεγχο των περιεκτών.

Ένα άλλο πολύ ενδιαφέρον χαρακτηριστικό, που ξεχωρίζει το Docker από το LXC, είναι ο περιορισμός του πλήθους των εφαρμογών που μπορούν να πακετοποιηθούν σε ένα περιέκτη. Το Docker βασίζεται στην ιδέα της πακετοποίησης μίας και μόνο εφαρμογής ανά περιέκτη, σε αντίθεση με το LXC στο οποίο πολλές εφαρμογές ή ακόμα και ολόκληρο λειτουργικό σύστημα πακετοποιούνται σε έναν περιέκτη. Αυτή η καινοτομία διευκόλυνε κατά κάποιο τρόπο τους προγραμματιστές, όμως από την άλλη τους ενέπλεξε σε μια διαδικασία εκμάθησης, ώστε να μπορούν να αναπτύσσουν μία εφαρμογή που να μπορεί να διανέμεται σε διάφορους περιέκτες. Ένα τέτοιο περιβάλλον απαιτεί σχολαστικό σχεδιασμό και αποφάσεις σε επίπεδο αρχιτεκτονικής και εντολές προγραμματισμού (scripting) για την υποστήριξη των εφαρμογών, γεγονός που περιπλέκει τη μετακόμιση εφαρμογών από φυσικούς εξυπηρετητές ή εικονικές μηχανές σε περιέκτες Docker, σε αντίθεση με τη μετακόμιση τους σε LXC περιέκτες.

2.4.1.2.1 Docker εργαλεία (tools used by Docker)

Η τεχνολογία Docker αξιοποιεί ένα ευρύ φάσμα εργαλείων, με τη Docker διεπαφή γραμμής εντολών (Docker CLI) να αποτελεί ένα από τα κύρια εργαλεία που περιλαμβάνει. Χρησιμοποιώντας αυτό το εργαλείο, γίνεται δυνατή η διαχείριση και η επεξεργασία των Docker εικόνων (images). Αυτές οι εικόνες, που μπορούν να αναζητηθούν στο δημόσιο αποθετήριο Docker Hub, χρησιμοποιούνται για τη δημιουργία του περιβάλλοντος των διαφόρων εφαρμογών. Σε αντίθεση με το LXC, του οποίου τα εργαλεία χρησιμοποιούνται ευρέως σε Linux περιβάλλοντα, τα Docker εργαλεία είναι ιδιοταγή, που σημαίνει ότι απαιτούν πιο εξειδικευμένη γνώση.

Το Docker Compose είναι ένα άλλο εργαλείο, το οποίο επιτρέπει την εκκίνηση εφαρμογών οι οποίες είναι διαχωρισμένες σε διαφορετικούς περιέκτες, ενώ παράλληλα επιτρέπει την ανταλλαγή δεδομένων μεταξύ αυτών. Επίσης το Docker υποστηρίζει τη διαχείριση συστοιχιών Docker περιεκτών, με το εργαλείο ενορχήστρωσης (orchestrator engine / orchestrator) που έχει αναπτύξει και ονομάζεται Docker Swarm.

Τέλος, το Docker προσφέρει και το Docker Trusted Registry, ένα ιδιωτικό αποθετήριο επαγγελματικής χρήσης, στο οποίο μπορούν να αποθηκευτούν πιστοποιημένες εικόνες Docker. Ωστόσο, για την χρήση αυτού του αποθετηρίου απαιτείται η επαγγελματική έκδοση του Docker (Docker EE).

2.4.1.2.2 Πλεονεκτήματα του Docker

Βάσει των προαναφερθέντων, είναι σαφές ότι η τεχνολογία Docker είναι αρκετά πιο εξελιγμένη από την αντίστοιχη LXC, από την οποία προήλθε. Το κύριο πλεονέκτημα είναι η διαθεσιμότητα μηχανής ενορχήστρωσης περιεκτών, χάρη στην οποία γίνεται δυνατή η διαχείριση μεγάλων συστοιχιών από περιέκτες, καθιστώντας το Docker ως μία πιο επεκτάσιμη τεχνολογία.

Ένα άλλο πλεονέκτημα, είναι η δυνατότητα διαχωρισμού μιας εφαρμογής σε διάφορα μικρότερα κομμάτια, τα οποία μπορούν να εκτελεστούν σε διαφορετικούς περιέκτες. Αυτή η προσέγγιση των μικροϋπηρεσιών (microservices), αυξάνει την ευελιξία και την ανεξαρτησία των διακριτών συνιστωσών της εφαρμογής, επιτρέποντάς της να ανταποκρίνεται με ευελιξία και να κλιμακώνεται ανάλογα με τις απαιτήσεις που προστάζει ο εκάστοτε φόρτος εργασιών.

Ένα επίσης σημαντικό πλεονέκτημα του Docker είναι η δυνατότητα του να εκτελείται και σε άλλα λειτουργικά συστήματα πέρα από τις Linux διανομές, όπως σε Windows, αλλά και η μεγάλη υποστήριξη που λαμβάνει από τους μεγαλύτερους παρόχους υπολογιστικής νέφους, όπως η Amazon, η Microsoft, η Google και η IBM. Αυτό σημαίνει ότι υπάρχει μεγάλη κοινότητα και πολλοί διαθέσιμοι πόροι για υποστήριξη, εκπαίδευση και ανάπτυξη σε περιβάλλοντα Docker.

Όλα τα παραπάνω πλεονεκτήματα καθιστούν την τεχνολογία Docker ένα ισχυρό εργαλείο για την ανάπτυξη και την διαχείριση εφαρμογών, παρέχοντας απομόνωση, ευελιξία και επεκτασιμότητα.

2.4.1.2.3 Μειονεκτήματα του Docker

Ένα από τα κυριότερα προβλήματα της τεχνολογίας Docker, η οποία αφορά κυρίως τις ομάδες ανάπτυξης λογισμικού, είναι η δυσκολία σχεδιασμού και ανάπτυξης των ίδιων των εφαρμογών. Το Docker καινοτόμησε πακετοποιώντας μόνο μια εφαρμογή ανά περιέκτη και διαχωρίζοντας την πλήρη εφαρμογή σε διακριτά κομμάτια, τα οποία αποθηκεύει σε διαφορετικούς περιέκτες. Λόγω αυτού οι εφαρμογές πρέπει να σχεδιάζονται ώστε να μπορούν να διανεμηθούν σε διαφορετικούς περιέκτες, γεγονός που απαιτεί εκτενέστερο σχεδιασμό, αφού πρόκειται για μία πιο πολύπλοκη αρχιτεκτονική, σε σχέση με την παραδοσιακή ανάπτυξη εφαρμογών.

Σύμφωνα με τα προαναφερθέντα, προκύπτει και το πρόβλημα της μεταφοράς μιας παραδοσιακής εφαρμογής σε Docker περιέκτες, η οποία μπορεί να μην είναι δυνατόν να πραγματοποιηθεί αν πρώτα δεν διαιρεθεί σε μικρότερα κομμάτια.

Επιπροσθέτως η διαχείριση συστοιχιών περιεκτών Docker είναι αρκετά πιο πολύπλοκη από την αντίστοιχη LXC, γεγονός που δημιουργεί την ανάγκη για μηχανές ενορχήστρωσης, οι οποίες διευκολύνουν την απαιτητική διαχείριση των όλο και πιο σύνθετων συστοιχιών [35].

2.4.1.3 Containerd

Το containerd είναι ένα περιβάλλον εκτέλεσης περιεκτών (container runtime) το οποίο σχεδιάστηκε με έμφαση στην απλότητα, την ευρωστία και την φορητότητα. Αρχικά δημιουργήθηκε από το Docker και στη συνέχεια υιοθετήθηκε από το CNCF.

Το containerd αναλαμβάνει τη διαχείριση του κύκλου ζωής ενός περιέκτη σε ένα φυσικό ή εικονικό μηχάνημα. Αποτελεί ένα υποσύνολο του αρχικού Docker Engine και αναλαμβάνει την εκτέλεση των περιεκτών, τη διαχείριση των εικόνων περιέκτη και τη διαχείριση του αποθηκευτικού χώρου. Είναι μία διεργασία δαίμονα η οποία δημιουργεί, ξεκινά, σταματά και καταστρέφει περιέκτες. Ωστόσο δεν εισχωρεί σε θέματα δικτύωσης .

Το containerd δεν προσφέρει χαρακτηριστικά (features) που να προορίζονται για χρήση από τους προγραμματιστές , αλλά αποτελεί ένα runtime εργαλείο κατάλληλο για χρήσεις σε υλοποιήσεις μεγάλης κλίμακας. Χρησιμοποιείται, συχνά, ως μέρος

της υποδομής ενός ενορχηστρωτή περιεκτών, όπως το Kubernetes, και φυσικά το Docker, για την εκτέλεση και τη διαχείριση των περιεκτών. [21]

2.4.1.4 CRI-O

Το CRI-O είναι ένα περιβάλλον εκτέλεσης (runtime) για περιέκτες, που δημιουργήθηκε από τη Red Hat, με σκοπό να παρέχει ένα πιο απλό και ελαφρύ runtime, που να πληροί τις απαιτήσεις του Kubernetes. Το CRI-O είναι σχεδιασμένο να είναι απλούστερο από άλλες λύσεις, θέλοντας να ακολουθήσει την φιλοσοφία των UNIX, που λέει «κάνε κάθε πρόγραμμα να κάνει ένα πράγμα καλά», εξ' ου και πρόκειται για μια μινιμαλιστική CRI διεπαφή.

Αρχικά, το CRI-O δημιουργήθηκε ως μία λύση για το OpenShift - την πλατφόρμα της RedHat για περιέκτες - αλλά στη συνέχεια υποστηρίχτηκε και από την SUSE και την Intel. Είναι συμβατό με τη CRI διεπαφή και με το τύπο format περιεκτών Docker και OCI. Επίσης υποστηρίζει διάφορα CoW (Copy-on-Write) συστήματα αρχείων (filesystems) όπως το aufs και το Btrfs.

Το CRI-O χρησιμοποιεί το runC για την εκκίνηση των περιεκτών και τις βιβλιοθήκες του skopeo για τη λήψη εικόνων περιεκτών και την δημιουργία των συστημάτων αρχείων των περιεκτών. Για την διαχείριση των περιεκτών χρησιμοποιεί ένα δαίμονα με την ονομασία conmon [36].

Ο συνδυασμός αυτών των εργαλείων και των βιβλιοθηκών επιτρέπει στο CRI-O να παρέχει μια απλή και ελαφριά εμπειρία εκτέλεσης περιεκτών, που ανταποκρίνεται στις ανάγκες του Kubernetes.

2.4.1.5 runC

Το runC είναι ένα εργαλείο για την εκκίνηση και την εκτέλεση των περιεκτών, που είναι συμβατό με το OCI. Σχεδιάστηκε με ιδιαίτερη έμφαση στην ασφάλεια και όπως το containerd, αρχικά, δημιουργήθηκε και αυτό από το Docker. Σήμερα, είναι ένα αυτόνομο εργαλείο, δεν εξαρτάται από το Docker και χρησιμοποιείται σε παραγωγικά περιβάλλοντα μεγάλης κλίμακας.

Ως standard runtime [30] [37] για περιέκτες, σύμφωνα με το OCI, το runC παρέχει τις βασικές λειτουργίες για τη διαχείριση περιεκτών, αλλά δεν υλοποιεί την προδιαγραφή για τις εικόνες περιέκτη του OCI. Για την υποστήριξη της

προδιαγραφής του OCI, υπάρχουν άλλα, υψηλού επιπέδου περιβάλλοντα εκτέλεσης, όπως το containerd που αναφέρθηκε προηγουμένως, τα οποία είναι υπεύθυνα για την επεξεργασία εικόνων περιέκτη και τη διαχείριση περιεκτών, τα οποία, περιβάλλοντα εκτέλεσης, χρησιμοποιούν το runC.

Τέλος, το runC υποστηρίζει πλήρως τα Linux namespaces και έχει ενδογενή υποστήριξη των χαρακτηριστικών ασφάλειας του Linux, όπως τα SELinux και το AppArmor.

2.4.2 Ενορχηστρωτές Περιεκτών

Η ανάγκη για ευελιξία στην επιχειρηματικότητα έχει οδηγήσει σε μια διαρκή πίεση για συχνότερη προσφορά και ανάπτυξη λογισμικού. Η ανάγκη αυτή ικανοποιείται με τεχνικές όπως η 'ευέλικτη ανάπτυξη εφαρμογών' (Agile), με DevOps (Development Operations) μεθόδους και εργαλεία που καθιστούν την ανάπτυξη λογισμικού έναν συνεχή κύκλο ζωής (continuous lifecycle) [38]. Παράλληλα με αυτή τη τάση έχει υιοθετηθεί και το αρχιτεκτονικό μοτίβο των μικροϋπηρεσιών, που ουσιαστικά, αποσυνθέτει τη μονολιθική δομή σε ανεξάρτητα στοιχεία, τα οποία προσφέρουν ευκολία στην ανάπτυξη, στην διαχείριση και στην οριζόντια κλιμάκωση (horizontal scaling). Εν συντομία, με το μοντέλο αυτό δημιουργούνται μεγάλα συστήματα εφαρμογών και υπηρεσιών, έχοντας ως δομικά στοιχεία τις μικροϋπηρεσίες. Κάθε εφαρμογή συντίθεται από ανεξάρτητες υπηρεσίες - μικροϋπηρεσίες - οι οποίες υλοποιούν συγκεκριμένες λειτουργίες και επικοινωνούν μέσω APIs.

Οι περιέκτες προσφέρουν ένα ιδανικό μέσο για την διανομή των μικροϋπηρεσιών, λόγω της μικρής υπερπληροφορίας και της τάχιστας εκκίνησης τους. Επιπροσθέτως, είναι κατάλληλοι για οριζόντια εξάπλωση και αυτό μπορεί να επιτευχθεί με την υλοποίηση πολλαπλών πανομοιότυπων περιεκτών. Έτσι, σύγχρονες πολύπλοκες εφαρμογές μπορεί να απαρτίζονται από εκατοντάδες, ακόμα και χιλιάδες, υπηρεσίες με πολλαπλές, δυνητικά περίπλοκες, αλληλεξαρτήσεις [39].

Σύμφωνα με αυτές τις νέες τάσεις σχεδιασμού και ανάπτυξης των εφαρμογών, η χρήση περιεκτών για μεγάλες εφαρμογές μπορεί να είναι δύσκολο να υλοποιηθεί χωρίς κάποια λύση διαχείρισης των πολλαπλών περιεκτών. Έτσι καθιερώθηκε η χρήση των ενορχηστρωτών περιέκτη. Οι ενορχηστρωτές περιέκτη αυτοματοποιούν

την διάθεση των περιεκτών και την διαχείριση αυτών αναλαμβάνοντας την διαχείριση πόρων, του συντονισμού και της επικοινωνίας μεταξύ των μικροϋπηρεσιών.

Οι εφαρμογές που βασίζονται σε μικροϋπηρεσίες συχνά υλοποιούνται σε συστοιχίες εκατοντάδων στιγμιότυπων περιεκτών. Όσο το αποτύπωμα των εφαρμογών μεγαλώνει, τόσο αυξάνει και η πολυπλοκότητα διαχείρισης της συστοιχίας των περιεκτών που τις ορίζουν. Για να επωφεληθούν τα μέγιστα από τα πλεονεκτήματα των περιεκτών, οι συστοιχίες πρέπει να είναι αξιόπιστες, ανεκτικές σε σφάλματα, συνεχώς διαθέσιμες, επεκτάσιμες και ενδεχομένως γεωγραφικά διασκορπισμένες. Η διαχείριση των συστοιχιών παρουσιάζει μια αυξητική πολυπλοκότητα, όσο αυξάνονται οι περιέκτες και οι κόμβοι που τους φιλοξενούν. Οι ενορχηστρωτές περιέκτη μπορούν γενικευμένα να οριστούν ως συστήματα τα οποία προσφέρουν επαγγελματικού επιπέδου υπηρεσίες για την ενσωμάτωση και την διαχείριση των περιεκτών σε μεγάλη κλίμακα. Οι ενορχηστρωτές απλοποιούν την διαχείριση των περιεκτών και παρέχουν μία διεπαφή όχι μόνο για τον ορισμό της αρχικής ανάπτυξης των περιεκτών αλλά και για την διαχείριση ενός συνόλου περιεκτών ως μία οντότητα στα πλαίσια διαθεσιμότητας, επεκτασιμότητας και δικτύωσης.

Η ενορχήστρωση των περιεκτών επιτρέπει στους παρόχους υπηρεσιών νέφους να ορίσουν το πώς θα υλοποιούν, θα παρακολουθούν και θα ελέγχουν δυναμικά την διαμόρφωση των εφαρμογών, που υλοποιούνται από μια πληθώρα περιεκτών στο νέφος. Η ενορχήστρωση περιεκτών ασχολείται με την διαχείριση, τη διάθεση, την εκτέλεση, και τη συντήρηση των περιεκτών όταν αυτοί είναι σε παραγωγική, είτε όχι, λειτουργία. Οι ενορχηστρωτές περιέκτη συνήθως προσφέρουν τα ακόλουθα κύρια χαρακτηριστικά: **έλεγχο πόρων**, προγραμματισμό - **scheduling** - που θα τρέξει ο κάθε περιέκτης, **εξισορρόπηση φόρτου**, **έλεγχο υγείας** των περιεκτών, **ανοχή σφαλμάτων** και **αυτόματη κλιμάκωση** [38].

Ο **έλεγχος πόρων** επιτρέπει τη δέσμευση συγκεκριμένου ποσοστού ΚΜΕ και Μνήμης RAM για έναν περιέκτη. Οι περιορισμοί στη χρήση πόρων μπορούν να συντελέσουν στη λήψη αποφάσεων για καταλληλότερη αξιοποίηση των πόρων από τους περιέκτες και για την μη υπερέκλυση αυτών. Πράγματι ενώ ένας περιέκτης μπορεί να χρησιμοποιήσει όλους τους διαθέσιμους πόρους που υπάρχουν στο υποβόσκων σύστημα, οι ενορχηστρωτές, με τη χρήση APIs, περιορίζουν το

ποσοστό ΚΜΕ και Μνήμης που εν τέλει θα χρησιμοποιήσουν, σύμφωνα με τα όρια που έχουν οριστεί στην υλοποίηση (π.χ. Kubernetes Manifest αρχείο) του περιέκτη.

Ο **προγραμματισμός** ορίζει την πολιτική για το πλήθος των περιεκτών που θα τοποθετηθούν σε συγκεκριμένους κόμβους την εκάστοτε χρονική στιγμή. Ο προγραμματισμός μπορεί να γίνει βάσει των περιορισμών στους πόρους ή την συγγένεια μεταξύ των κόμβων ή βάσει και των δύο. Πιο εξελιγμένοι «προγραμματιστές» μπορούν να ενσωματωθούν σαν εξωτερικά στοιχεία.

Ο **εξισορροπητής φόρτου** αναλαμβάνει το έργο της κατανομής του φόρτου ανάμεσα στα διάφορα στιγμιότυπα περιεκτών. Ο πιο συνήθης υλοποιούμενος αλγόριθμος είναι ο Round-Robin. Πιο εξελιγμένες πολιτικές εξισορρόπησης φορτίου μπορούν πάντα να προσφερθούν από εξωτερικούς εξισορροπητές φορτίων.

Ο **έλεγχος υγείας** επιτυγχάνεται ελέγχοντας αν ένας περιέκτης είναι ικανός στο να απαντάει σε αιτήματα, σε συνήθη πρωτόκολλα όπως TCP, UDP, SSH και HTTP.

Η **ανοχή σφαλμάτων** μπορεί να υλοποιηθεί ως μια τεχνική ελέγχου των αντιγράφων - replicas - των περιεκτών. Ο έλεγχος των αντιγράφων επιτρέπει τον ορισμό και την διατήρηση του επιθυμητού αριθμού περιεκτών που θέλουμε. Σε συνδυασμό με τον έλεγχο υγείας χρησιμοποιείται για να καθορίσει πότε ένας ελλαττωματικός περιέκτης πρέπει να καταστραφεί για να εκκινήσει ένας νέος στη «θέση» του, έτσι ώστε να διατηρηθεί ο επιθυμητός αριθμός υγιών αντιγράφων.

Η **αυτόματη κλιμάκωση** επιτρέπει την αυτόματη προσθήκη ή αφαίρεση περιεκτών. Οι υλοποιούμενες πολιτικές βασίζονται σε κάποια όρια (ΚΜΕ και μνήμης), αλλά σε κάποιες περιπτώσεις είναι πιθανό να οριστούν πιο εξειδικευμένες πολιτικές αυτόματης κλιμάκωσης, που να βασίζονται σε διαφορετικές παραμέτρους.

Στην περίπτωση των ενορχηστρωτών περιεκτών υπάρχουν αρκετές λύσεις. Οι διαφορές μεταξύ των ενορχηστρωτών βασίζονται στην συμπεριφορά και στις λύσεις διαχείρισης που προσφέρουν. Δεν είναι όλοι ικανοί να ανταπεξέλθουν στις απαιτήσεις που θέτουν τα παραγωγικά περιβάλλοντα των μεγάλων υποδομών και δεν έχουν όλοι τις ίδιες δυνατότητες. Το πλήθος των περιεκτών το οποίο μπορεί να διαχειριστεί ένας ενορχηστρωτής είναι καθοριστικό για την επιλογή του, από την εκάστοτε τεχνολογία που είναι επιθυμητό να υλοποιηθεί.

Συμπληρωματικά με του ενορχηστρωτές περιεκτών υπάρχουν και οι πλατφόρμες περιεκτών, οι οποίες ουσιαστικά προσφέρουν ένα επιπλέον επίπεδο αφάιρεσης (abstraction) σε σχέση με τους ενορχηστρωτές με στόχο την αποτελεσματικότερη, φιλικότερη και απλούστερη διαχείριση των ενορχηστρωτών. Ακολουθεί μια συνοπτική αναφορά στους πέντε πιο χρησιμοποιούμενους ενορχηστρωτές περιέκτη.

2.4.2.1.1 Docker Compose

Το Docker Compose είναι μία λύση η οποία είναι κατάλληλη για εφαρμογές που απαρτίζονται από πολλαπλούς Docker περιέκτες. Αυτή η λύση βασίζεται σε αρχεία διαμόρφωσης YAML, στα οποία ορίζονται οι διασυνδέσεις των περιεκτών και άλλα χαρακτηριστικά όπως εικόνες και χώροι αποθήκευσης.

Το πλεονέκτημα αυτής της λύσης είναι ότι είναι απλή στην υλοποίηση, στην αναγνωσιμότητα και στην διαφάνεια της διαμόρφωσης της. Ορίζοντας τις απαιτούμενες υπηρεσίες, τις εξαρτήσεις τους και τις ρυθμίσεις τους σε ένα αρχείο Docker Compose, το περιβάλλον της εφαρμογής, μπορεί να αναπαραχθεί σε οποιοδήποτε μηχάνημα έχει εγκατεστημένο το Docker Compose.

Στα μειονεκτήματα συγκαταλέγονται οι περιορισμένες δυνατότητες σε σχέση με άλλες λύσεις και η απουσία χαρακτηριστικών όπως η υψηλή διαθεσιμότητα και η κατανομή περιεκτών σε συστοιχίες. Το Docker Compose μπορεί να θεωρηθεί κατάλληλο για ανάπτυξη, αλλά όχι για μεγάλες παραγωγικές υποδομές.

2.4.2.1.2 Docker Swarm

Το Docker Swarm είναι μία λύση που ομοιάζει αρκετά με το Kubernetes όσον αφορά την αρχιτεκτονική και τα χαρακτηριστικά του. Υπάρχει δυνατότητα για κυλιόμενες ενημερώσεις (rolling updates), αποκεντρωμένη σχεδίαση, δικτύωση πολλαπλών κόμβων, εξισορρόπηση φόρτου και επίσης υποστηρίζει εγγενώς τα Compose αρχεία διαμόρφωσης.

Σε σύγκριση με το Docker Compose μπορεί να θεωρηθεί ως μια πιο εξελιγμένη λύση, η οποία προσφέρει αρκετές ενσωματωμένες δυνατότητες. Παρόλα αυτά, οι δυνατότητες του εξακολουθούν να υστερούν σε σχέση με το μέγεθος της κοινότητας και το οικοσύστημα γύρω από το Kubernetes. Η λύση Swarm είναι κατάλληλη για μικρές συστοιχίες, έχει απλή διαχείριση και ανάπτυξη και χρησιμοποιείται σε μικρά

παραγωγικά περιβάλλοντα. Το κύριο μειονέκτημα του Swarm είναι ότι δουλεύει μόνο με Docker περιέκτες και δεν υποστηρίζει αυτόματη κλιμάκωση.

2.4.2.1.3 Apache Mesos

Το Apache Mesos ξεκίνησε να αναπτύσσεται το 2009 από το Πανεπιστήμιο της Καλιφόρνια στο Μπέρκλεϊ, ως ένα εργαλείο για την διαχείριση συστοιχιών [40], ανεξάρτητα από τους περιέκτες. Η διαφορά σε σύγκριση με άλλες λύσεις είναι ότι το Mesos δεν προορίζεται μόνο για ενορχήστρωση περιεκτών, αλλά και για ενορχήστρωση διαφορετικών φόρτων εργασίας. Είναι μια λύση, του Apache Foundation, που ασχολείται με διάφορα έργα (projects) όπως το Hadoop, το Spark, το Aurora κ.α..

Το Mesos χρησιμοποιεί μία αρχιτεκτονική master-agent για τη διαχείριση κατανεμημένων υπολογιστικών πόρων. Σε αυτήν την αρχιτεκτονική, υπάρχουν δύο τύποι κόμβων, ο master και ο agent. Οι κόμβοι master είναι υπεύθυνοι για τον συντονισμό και την διαχείριση των πόρων. Οι κόμβοι agent είναι υπεύθυνοι για την παροχή των πόρων και την εκτέλεση των εργασιών. Κάθε κόμβος agent είναι συνδεδεμένος με τον κόμβο master και αναφέρει τη διαθεσιμότητα των πόρων του, όπως επεξεργαστική ισχύ, μνήμη και αποθηκευτικό χώρο. Όταν ο master θέλει να εκτελεστεί κάποια εργασία, την αναθέτει σε έναν κόμβο agent, ο οποίος αναλαμβάνει την εκτέλεση της.

Με την βοήθεια του Chronos και του Marathon, το Mesos μπορεί να διαχειρίζεται και να εκτελεί Docker περιέκτες [41]. Το Chronos αναλαμβάνει τον προγραμματισμό εργασιών (job scheduler) σε συγκεκριμένο χρονικό πλαίσιο, ενώ το Marathon εστιάζει στη διαχείριση και την εκτέλεση εφαρμογών στο Mesos. Το Mesos συχνά συγκρίνεται με την ανοιχτού κώδικα λύση υπολογιστικού νέφους OpenStack και συνήθως χρησιμοποιείται για Big Data υπηρεσίες.

2.4.2.1.4 HashiCorp Nomad

Το Nomad είναι ένας ευέλικτος ενορχηστρωτής φόρτου εργασίας, που δίνει τη δυνατότητα σε έναν οργανισμό να αναπτύσσει και να διαχειρίζεται εύκολα οποιαδήποτε εφαρμογή χρησιμοποιώντας μια ενιαία, ενοποιημένη ροή εργασιών. Μπορεί να εκτελέσει διάφορους φόρτους εργασίας όπως περιέκτες, εικονικές μηχανές, παλιές εφαρμογές (legacy applications), μικροϋπηρεσίες και batch εφαρμογές. Το Nomad επιτρέπει στους προγραμματιστές να χρησιμοποιούν

δηλωτική υποδομή ως κώδικα (declarative Infrastructure as code) για την ανάπτυξη των εφαρμογών και χρησιμοποιεί “bin packing” για τον αποτελεσματικό προγραμματισμό εργασιών και τη βελτιστοποίηση της χρήσης των πόρων.

Στο Nomad, ορίζεται μια προδιαγραφή εργασίας, που περιγράφει την εφαρμογή ή την υπηρεσία που είναι επιθυμητό να αναπτυχθεί. Αυτή η προδιαγραφή είναι γραμμένη σε HashiCorp Configuration Language (HCL), η οποία είναι μια δηλωτική γλώσσα διαμόρφωσης (declarative configuration language) που χρησιμοποιείται από διάφορα εργαλεία της HashiCorp.

Ενώ οι σχετικές λύσεις όπως το Kubernetes επικεντρώνονται στη φιλοξενία εφαρμογών με περιέκτες, το Nomad, χρησιμοποιώντας την τεχνολογία εξομοίωσης QEMU [42], έχει την ικανότητα να εξυπηρετεί και εικονικές εφαρμογές, κάτι το οποίο δεν υποστηρίζεται εγγενώς από το Kubernetes, πάρα μόνο με την χρήση του kubvirt, και αυτό μπορεί να είναι ένα μεγάλο πλεονέκτημα σε ένα περιβάλλον με ετερογενείς φόρτους εργασίας. [43]

2.4.2.1.5 Kubernetes

Το Kubernetes είναι ένα πρόγραμμα ανοιχτού κώδικα για την αυτοματοποίηση της ανάπτυξης, της επεκτασιμότητας και της διαχείρισης εφαρμογών με περιέκτες. Έχει δημιουργηθεί από την εταιρία Google, η οποία αξιοποίησε εμπειρία, τουλάχιστον 10 χρόνων, εκτέλεσης περιεκτών σε παραγωγικά περιβάλλοντα. Η Google, στα παραγωγικά περιβάλλοντα της, χρησιμοποιεί αποκλειστικά τεχνολογίες περιέκτη [44]. Το Kubernetes είναι ένας προηγμένος εννοχηστρωτής που παρέχει υποστήριξη για διάφορες λύσεις περιέκτη, όπως τα Docker, rkt, cri-o, Windows Containers κ.α. Με την αρθρωτή του δομή και την ποικιλία εργαλείων που παρέχει, το Kubernetes επιτρέπει την επέκταση και προσαρμογή των δυνατοτήτων του ανάλογα με τις εκάστοτε ανάγκες.

Το Kubernetes βασίζεται σε μια αρχιτεκτονική Master-Worker. Οι Master κόμβοι λειτουργούν ως ο κεντρικός ελεγκτής και αναλαμβάνουν τον έλεγχο του συστήματος, συμπεριλαμβανομένης της διαχείρισης των Worker κόμβων. Οι κόμβοι Workers αποτελούν την εκτελεστική μονάδα, λαμβάνουν αιτήματα από τους Master κόμβους και εκτελούν τις αντίστοιχες εργασίες που τους έχουν ανατεθεί.

Μερικά από τα κύρια πλεονεκτήματα του Kubernetes είναι η δυνατότητά για αυτόματη κλιμάκωση, η αντιμετώπιση σφαλμάτων και η εξισορρόπηση του φόρτου

εργασίας. Επιπλέον, παρέχει ευελιξία στη διαχείριση των εφαρμογών και έχει τη δυνατότητα να αναβαθμίσει ή να επαναφέρει – σε περίπτωση αστοχίας - κάποια εφαρμογή, χωρίς να διακοπεί η λειτουργία της. Με την συνεχή ανάπτυξη και την ευρεία κοινότητα υποστήριξης, το Kubernetes παραμένει μια αξιόπιστη επιλογή για την οργάνωση και την διανομή εφαρμογών σε κλιμακώσιμα περιβάλλοντα.

Για την εκκίνηση και την διαχείριση των περιεκτών χρησιμοποιεί τρία βασικά στοιχεία. Το **Pod**, τη μικρότερη υλοποιήσιμη υπολογιστική μονάδα, στην οποία εκτελούνται ένας ή περισσότεροι περιέκτες. Μία εφαρμογή, η οποία αποτελείται από πολλούς περιέκτες, μπορεί να τους ενσωματώσει όλους σε ένα pod. Τον **Ελεγκτή Αντιγράφων** (Replication Controllers), ο οποίος εποπτεύει τα pods και διασφαλίζει ότι ένας συγκεκριμένος αριθμός Pod αντιγράφων εκτελείται ανά πάσα στιγμή. Το **Kubelet**, που εκτελείται σε όλους τους worker κόμβους της συστοιχίας και διαβάζει τα μανιφέστα των περιεκτών (container manifests) διασφαλίζοντας ότι έχουν ξεκινήσει και εκτελούνται οι περιέκτες που ορίζονται σε αυτά, και συνδυαστικά με άλλες υπηρεσίες, που εκτελούνται σε κάθε pod, διαχειρίζονται και εποπτεύουν τους περιέκτες. Το Kubelet εκτελείται και στους master κόμβους αλλά η κύρια του αποστολή είναι η διαχείριση των pods και των περιεκτών στους worker κόμβους.

2.4.3 Πλατφόρμες Περιεκτών

Οι Πλατφόρμες Περιεκτών (Container Platforms) δεν πρέπει να συγχέονται με τις Μηχανές Ενορχήστρωσης Περιεκτών (Container Orchestration Engines). Οι πλατφόρμες περιεκτών, μεταξύ άλλων, είναι υπεύθυνες για το πακετάρισμα της εφαρμογής σε ένα περιέκτη και για την επικείμενη διανομή της. Όμως, η διαδικασία αυτή δεν λύνει το πρόβλημα παρακολούθησης και απλούστερης διαχείρισης των περιεκτών, όσο εκτελείται η – πακετοποιημένη - εφαρμογή, που βρίσκεται εντός τους. Για αυτό το λόγο, υπάρχει η ανάγκη για ενορχηστρωτές περιεκτών οι οποίοι πέρα από τη παρακολούθηση των περιεκτών, υποστηρίζουν την δημιουργία, την εκκίνηση και την απόρριψη αυτών, καθώς και την εκκίνηση της εφαρμογής, την ενημέρωση αυτής, χωρίς να διακόπτεται η λειτουργία των υπηρεσιών, την παρακολούθηση της κατάστασης της και την επανεκκίνηση της, εάν διαπιστωθεί κάποιο σφάλμα.

Στη σφαίρα του Δημόσιου Νέφους, οι διάφοροι πάροχοι προσφέρουν, ο καθένας, την δικιά του εκδοχή πλατφόρμας περιεκτών για την διαχείριση της αντίστοιχης

λύσης ενορχήστρωσης περιεκτών που διαθέτουν. Η Amazon προσφέρει το Amazon Elastic Compute Cloud (EC2), η Google το Google Compute Engine (GCE), η Microsoft το Azure Kubernetes Service (AKS) καθώς και οι υπόλοιποι πάροχοι τις ιδιοταγείς λύσεις τους. Πέρα από τα Δημόσια Νέφη, υπάρχουν λύσεις που δεν περιορίζονται αποκλειστικά για αυτά, αλλά κυρίως για τα υπόλοιπα μοντέλα ανάπτυξης, τα Ιδιωτικά, τα Κοινοτικά ή τα Υβριδικά Νέφη. Τέτοιες λύσεις είναι το OpenStack, το OpenShift, το Rancher, το Nomad, το Portainer κ.α. που αποτελούν πλατφόρμες ανοιχτού κώδικα και επιτρέπουν την εύκολη ανάπτυξη, την ενορχήστρωση και τη διαχείριση περιβαλλόντων περιέκτη, ανεξάρτητα από την υποκείμενη υποδομή. [45] [46]

ΚΕΦΑΛΑΙΟ 3

Μετασχηματισμός Εφαρμογών και Υποδομών στα Σύγχρονα Πρότυπα Περιεκτών

Η διαρκώς αυξανόμενη τάση χρήσης των νεφούπολογιστικών υποδομών ([47], [48]) σε συνδυασμό με την επικρατούσα cloud native αρχιτεκτονική ανάπτυξης εφαρμογών, η οποία βασίζεται κυρίως στην χρήση μικρό-υπηρεσιών [49], καθώς και οι πρακτικές της συνεχούς ενοποίησης, παράδοσης και ανάπτυξης (CI/CD) συμπαρασεύρουν και επηρεάζουν την εξέλιξη των υπαρχόντων υποδομών και εφαρμογών. Οι εγκατεστημένες και λειτουργικές υποδομές οργανισμών και επιχειρήσεων στα πλαίσια της τακτικής συντήρησης ή αναβάθμισης τους οδηγούνται σε αναγκαίο μετασχηματισμό ώστε είτε οι εφαρμογές να μπορούν να φιλοξενηθούν με βέλτιστο τρόπο στις σύγχρονες νεφούπολογιστικές υποδομές, είτε οι υποδομές να έχουν την δυνατότητα να φιλοξενήσουν σύγχρονες εφαρμογές μικρο-υπηρεσιών. Η ανάγκη αυτή του μετασχηματισμού όπως περιεγράφηκε μπορεί με σαφήνεια να διαχωριστεί σε δύο βασικούς τύπους μετασχηματισμού, των εφαρμογών και των υποδομών φιλοξενίας.

3.1 Μετασχηματισμός υποδομής φιλοξενίας εικονικών μηχανών σε υποδομή συνφιλοξενίας περιεκτών

Σύμφωνα με στατιστικές οργανισμών και εταιριών που απασχολούνται με την ανάπτυξη και διαχείριση υπηρεσιών νέφους προκύπτει πως η χρήση τους αυξάνεται συνολικά, ανεξάρτητα από τον τύπο της υπηρεσίας (IaaS, PaaS, SaaS, CaaS κ.ο.κ.) [48] όσο και ανεξάρτητα από τον τρόπο διάθεσης τους (Public, Private, Hybrid Clouds) ενώ παράλληλα οι περισσότερες υπηρεσίες νέφους είναι χτισμένες πάνω σε πλατφόρμες Infrastructure as a Service (IaaS). Η IaaS υπηρεσία παρέχει τη θεμελιώδη υποδομή που επιτρέπει στους οργανισμούς να δημιουργούν και να αναπτύσσουν τις δικές τους εφαρμογές και υπηρεσίες στο cloud. Παρέχοντας εικονικούς υπολογιστικούς πόρους, όπως διακομιστές, αποθήκευση και δικτύωση, το IaaS επιτρέπει στους χρήστες να παρέχουν και να διαχειρίζονται εύκολα τους πόρους που χρειάζονται για την εκτέλεση των εφαρμογών τους χωρίς να χρειάζεται να ανησυχούν για την υποκείμενη φυσική υποδομή. Αυτό παρέχει μια εξαιρετικά ευέλικτη και επεκτάσιμη πλατφόρμα που μπορεί να προσαρμοστεί για να καλύψει

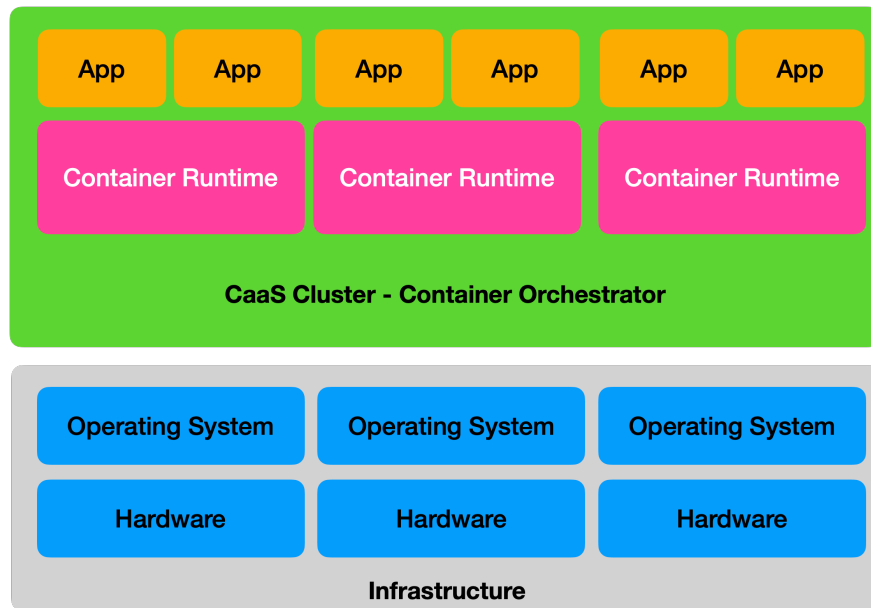
τις συγκεκριμένες ανάγκες διαφορετικών εφαρμογών και φόρτου εργασίας. Έτσι τα περισσότερα μοντέλα υπηρεσιών cloud, όπως το Platform as a Service (PaaS) και το Software as a Service (SaaS) ή το Container as a Service, βασίζονται πάνω σε IaaS υποδομή για να παρέχουν υπηρεσίες και λειτουργικότητα υψηλότερου επιπέδου.

Για την προσθήκη της δυνατότητας φιλοξενίας περιεκτών παράλληλα με τη φιλοξενία εικονικών μηχανών υπάρχουν τρεις βασικές προσεγγίσεις ([50], [51]).

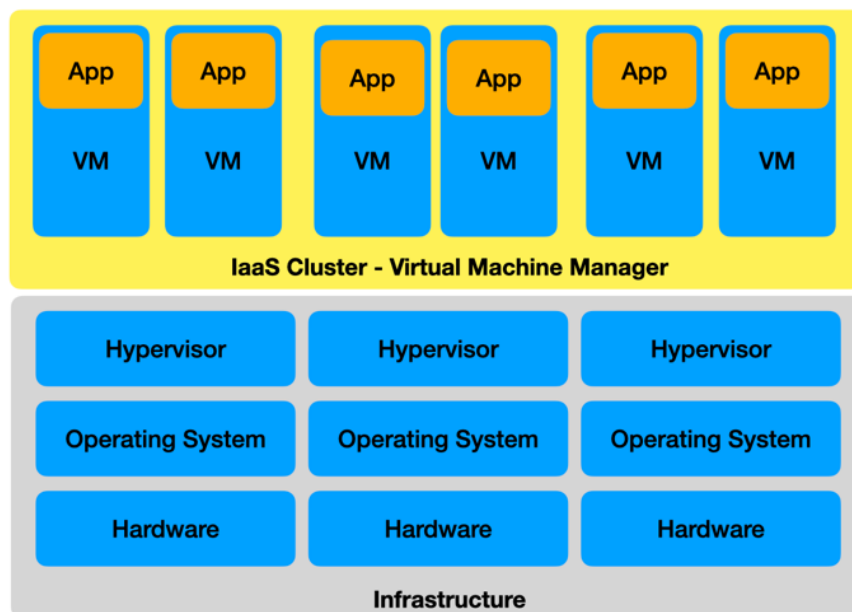
3.1.1 Παράλληλη υποδομή (Side to side)

Η πρώτη προσέγγιση στηρίζεται στην ανάπτυξη, εντός του ίδιου κέντρου δεδομένων, μιας δεύτερης ανεξάρτητης υποδομής (standalone CAAS) η οποία θα φιλοξενεί αποκλειστικά περιέκτες και θα προσφέρει υπηρεσίες φιλοξενίας περιεκτών. Η προσέγγιση αυτή ξεχωρίζει για την απλότητα της, τη βελτιστοποιημένη απόδοση [52] και την στοχευμένη λειτουργία της, αλλά επιβαρύνεται με ένα σημαντικό μειονέκτημα. Απαιτεί την προσθήκη / προμήθεια πρόσθετου εξοπλισμού που θα αξιοποιηθεί για τον σκοπό αυτό και ταυτόχρονα δεσμεύει τους πόρους αυτούς σε αποκλειστική χρήση αυξάνοντας τον κίνδυνο μειωμένης απόδοσης, υψηλού κόστους και χαμηλής ευελιξίας [53]. Πιο συγκεκριμένα, εάν η διαστασιολόγηση της υποδομής δεν υλοποιηθεί με βέλτιστο τρόπο ή οι ανάγκες μεταβάλλονται διαρκώς υπάρχει περίπτωση να δεσμεύονται φυσικοί πόροι (π.χ. εξυπηρετητές) για την αποκλειστική φιλοξενία περιεκτών ενώ προκύπτουν νέες ανάγκες για φιλοξενία εικονικών μηχανών ή και αντίστροφα. Το μειονέκτημα αυτό μπορεί να απαλύνεται όταν αναφερόμαστε σε υποδομές ιδιαίτερα μεγάλου μεγέθους, όπως σε υποδομές παρόχων δημόσιου νέφους, όπου τα διαχειριστικά ή λειτουργικά κόστη (OPEX) βελτιστοποιούνται υλοποιώντας ανεξάρτητες υποδομές και απορροφούν τα κόστη απόκτησης εξοπλισμού (CAPEX). Στις περιπτώσεις όπου αναφερόμαστε σε ιδιωτικά νέφη μικρών ή μεσαίων επιχειρήσεων και οργανισμών η δέσμευση πόρων για αποκλειστική χρήση πρέπει να πραγματοποιείται μετά από ενδελεχή μελέτη και τεκμηρίωση. Στην προσέγγιση αυτή μπορούν να αξιοποιηθούν οι περισσότερες λύσεις φιλοξενίας και ενορχήστρωσης περιεκτών με δημοφιλέστερες τις λύσεις που βασίζονται σε

συστοιχίες διανομών Kubernetes. Μερικές απ' αυτές περιλαμβάνουν το Kubernetes, το OpenShift, το Docker Swarm, το Apache Mesos.



Σχήμα 3. Απεικόνιση αρχιτεκτονικής συστοιχίας ενορχήστρωσης περιεκτών

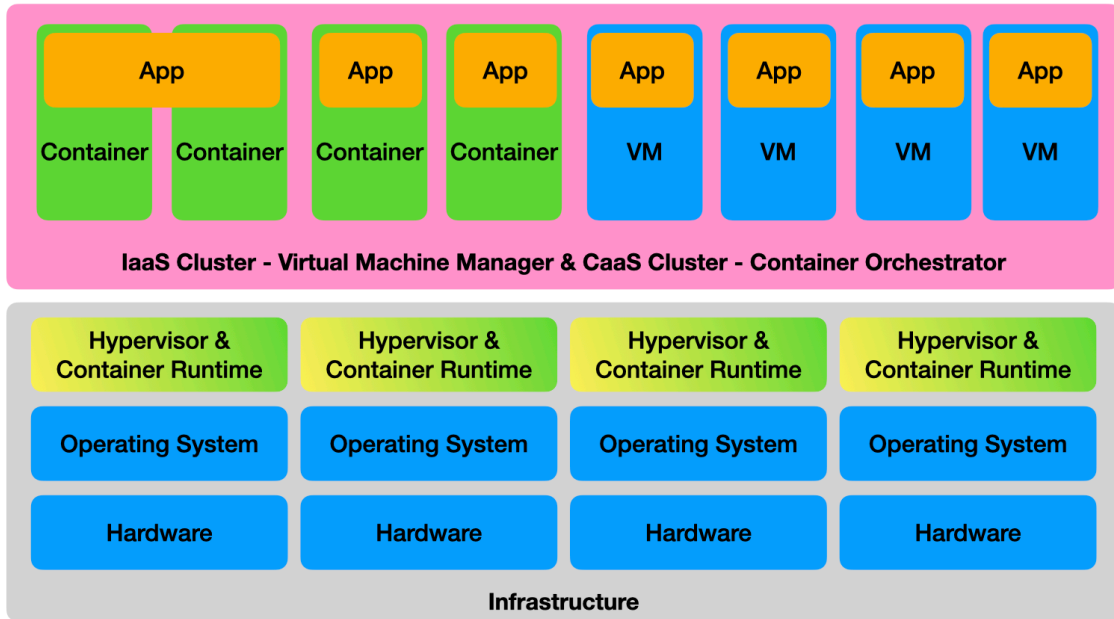


Σχήμα 4. Απεικόνιση αρχιτεκτονικής συστοιχίας διαχειριστή εικονικών μηχανών

3.1.2 Ενιαία υποδομή (All in One)

Η δεύτερη προσέγγιση αφορά την ανάπτυξη και υλοποίηση ενιαίας υποδομής που θα παρέχει εγγενώς δυνατότητες ταυτόχρονης φιλοξενίας εικονικών μηχανών και περιεκτών [54]. Στην προσέγγιση αυτή, η οποία εξορισμού της αποτελεί

φαινομενικά την ιδανικότερη λύση, κρίσιμος παράγοντας είναι η εύρεση και επιλογή του κατάλληλου λογισμικού / πλατφόρμας που θα ενσωματώνει τις απαραίτητες λειτουργίες και δυνατότητες. Πρακτικά, για την επιτυχή υλοποίηση της προσέγγισης αυτής θα πρέπει να αξιοποιηθεί λογισμικό που θα καλύπτει τόσο τις δυνατότητες ενός διαχειριστή εικονικών μηχανών (Virtual Machine Manager), όσο και μιας πλατφόρμας ενορχήστρωσης περιεκτών (Orchestration Platform). Παράλληλα, θα πρέπει να παρέχει και όλες τις απαιτούμενες δυνατότητες διαχείρισης πόρων, επεκτασιμότητας, υψηλής διαθεσιμότητας και αξιοπιστίας που κρίνονται απαραίτητες στην διάθεση υπηρεσιών νέφους. Η προσέγγιση αυτή μεγιστοποιεί την αποδοτικότητα αφού όλος ο εξοπλισμός μπορεί να αξιοποιηθεί από την πλατφόρμα για οποιουδήποτε τύπου υπηρεσία απαιτηθεί να φιλοξενηθεί και παράλληλα μειώνεται το κόστος απόκτησης και διαχείρισης αφού ο φυσικός εξοπλισμός που απαιτείται αναμένεται να είναι ίδιος ή μειωμένος συγκριτικά με την πρώτη προσέγγιση και οι διαχειριστές καλούνται να συντηρήσουν μία ενιαία πλατφόρμα. Η δυσκολία που υποκρύπτει η συγκεκριμένη προσέγγιση έγκειται στην εξεύρεση της κατάλληλης λύσης που θα πληροί τα κριτήρια που θέτει η επιχείρηση / οργανισμός αλλά και στην διαχείριση / συντήρηση της υποδομής αυτής δεδομένου ότι οι λύσεις που ενσωματώνουν τις προαναφερθείσες δυνατότητες χαρακτηρίζονται ως πολύπλοκες και απαιτούν υψηλό επίπεδο τεχνογνωσίας. Λύσεις που εν γένει ικανοποιούν την προσέγγιση αυτή αποτελούν το OpenStack [55] και το TrueNAS SCALE, ενώ λύσεις όπως το Open Nebula και το Proxmox που υποστηρίζουν LXC περιέκτες δεν μπορούν να θεωρηθούν πλήρεις αφού η πληθώρα των περιεκτών που χρησιμοποιούνται είναι τύπου Docker.

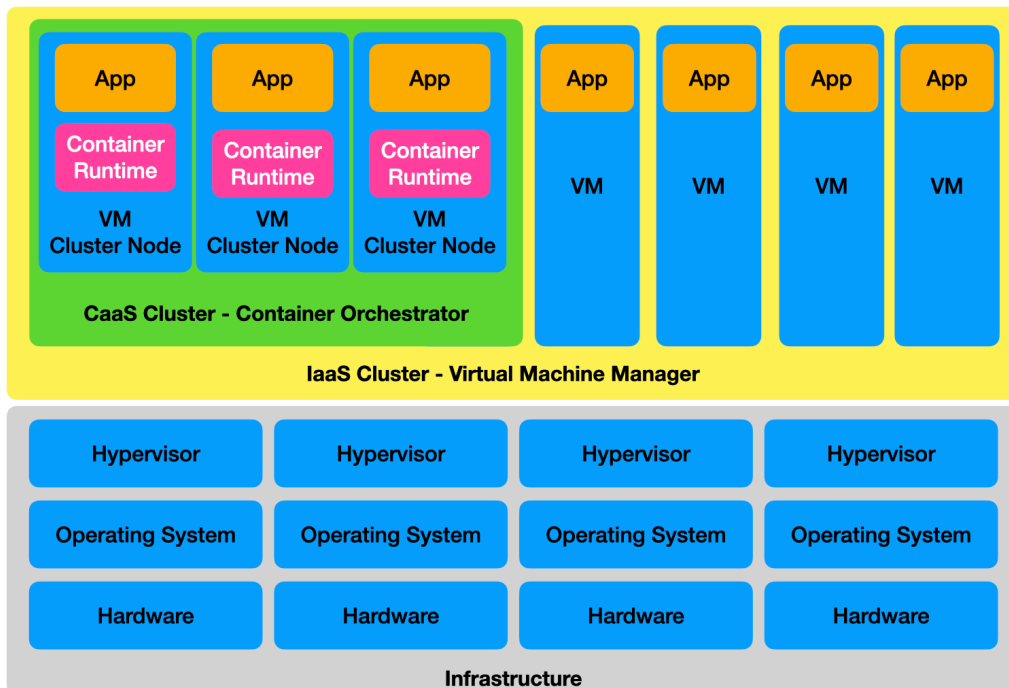


Σχήμα 5. Απεικόνιση αρχιτεκτονικής συστοιχίας ενιαίας πλατφόρμας διαχείρισης εικονικών μηχανών και περιεκτών

3.1.3 Υβριδική υποδομή (Hybrid - CaaS over IaaS)

Η τρίτη προσέγγιση αποτελεί μια υβριδική λύση και στηρίζεται στην υλοποίηση δύο "εικονικά" ανεξάρτητων υποδομών με τη διαφορά ότι η υποδομή διαχείρισης και ενορχήστρωσης περιεκτών τοποθετείται από άποψη αρχιτεκτονικής ένα επίπεδο ψηλότερα από την υποδομή διαχείρισης εικονικών μηχανών. Πιο αναλυτικά, υλοποιείται η τεχνική των εικονικών συστοιχιών (virtual clusters) και η υποδομή φιλοξενίας εικονικών μηχανών παραμένει ως η κυρίαρχη υποδομή διάθεσης πόρων. Οι πόροι αυτοί (εικονικές μηχανές κ.λπ.) αξιοποιούνται από την υποδομή διαχείρισης περιεκτών για την φιλοξενία της. Αυτό σημαίνει ότι αντί η υποδομή φιλοξενίας περιεκτών να εγκατασταθεί σε φυσικό εξοπλισμό, ανεξάρτητο της IaaS υποδομής που παρέχει τις εικονικές μηχανές, εγκαθίσταται σε εξυπηρετητές / εικονικές μηχανές που παρέχονται από την υποδομή IaaS και υλοποιούν μία ή περισσότερες εικονικές συστοιχίες. Η προσέγγιση αυτή, αν και περιγράφεται ίσως ως η πιο περίπλοκη αρχιτεκτονικά, προσφέρει σημαντικά πλεονεκτήματα έναντι των δύο προηγούμενων. Με την επιλογή της προσέγγισης αυτής δεν απαιτείται η εκ νέου υλοποίηση υποδομής εικονικών μηχανών (IaaS) και παράλληλα απαιτείται μικρότερη ή και καθόλου προσθήκη νέου εξοπλισμού συγκριτικά με την πρώτη προσέγγιση. Παράλληλα, η διατήρηση της υποδομής IaaS δεν τροποποιεί τα διαχειριστικά κόστη αφού η υποδομή παραμένει η ίδια, αλλά

παράλληλα διευρύνει τις επιλογές πλατφόρμας ενορχήστρωσης περιεκτών, δεδομένου ότι οι περισσότερες λύσεις μπορούν να εγκατασταθούν είτε σε φυσικές είτε σε εικονικές μηχανές. Επιπλέον η διαχείριση και διάθεση των πόρων (αποθηκευτικά συστήματα, δίκτυα, κ.α.) εξακολουθεί να παρέχεται μέσω της IaaS υποδομής μειώνοντας ζητήματα διαχείρισης ή διαλειτουργικότητας και εξασφαλίζει την υψηλή διαθεσιμότητα τους μέσω των δυνατοτήτων που διαθέτει η IaaS υποδομή. Μειονέκτημα της προσέγγισης αυτής μπορεί να θεωρηθεί ο επιπλέον φόρτος (overhead) που προκύπτει από την προσθήκη ενός ακόμα επιπέδου στην αρχιτεκτονική της προσέγγισης αναφορικά με την πλατφόρμα ενορχήστρωσης. Λόγω όμως των συνεχών νέων τεχνικών βελτιστοποίησης [56], ως προς την κατανάλωση πόρων, τόσο των hypervisors και των διαχειριστών εικονικών μηχανών (VMMs) όσο και των ενορχηστρωτών περιεκτών (Orchestrators) αλλά και των υψηλών επιδόσεων του φυσικού εξοπλισμού, η επιβάρυνση που προκύπτει, αν και δεδομένη, μπορεί να θεωρηθεί αμελητέα ή τουλάχιστον ανεκτή σε όρους παραγωγικών συστημάτων.



Σχήμα 6. Απεικόνιση αρχιτεκτονικής συστοιχίας ενορχηστρωτή περιεκτών σε συστοιχία διαχείρισης εικονικών μηχανών

Μια ακόμα προσέγγιση, που δεν περιγράφεται ως ξεχωριστή, αλλά θα μπορούσε να πλαισιωθεί στην δεύτερη προσέγγιση που αφορά την αξιοποίηση μιας ενιαίας

πλατφόρμας αφορά την αξιοποίηση πλατφορμών ενορχήστρωσης περιεκτών με ενσωμάτωση εργαλείων (επεκτάσεις) που επιτρέπουν την εκτέλεση και διαχείριση εικονικών μηχανών. Οι λύσεις αυτές (KubeVirt, Virtlet), αν και πλέον διαθέτουν επαρκείς δυνατότητες, δεν μπορούν να συγκριθούν για την σταθερότητα και αξιοπιστία που απαιτείται και προσφέρουν οι διαχειριστές εικονικών μηχανών (VMMs), ενώ συνήθως δεν διαθέτουν επαρκή τεκμηρίωση και υποστήριξη ακόμα και αν προσφέρονται σε σταθερές εκδόσεις. Είναι, όμως, σχεδόν βέβαιο πως οι μελλοντικές εκδόσεις θα αποτελέσουν μια τέταρτη προσέγγιση ή θα ενσωματωθούν ως λύσεις στην δεύτερη προσέγγιση.

Με την εκτίμηση πως και οι τρεις προσεγγίσεις που περιεγράφηκαν μπορούν να αξιολογηθούν και να χαρακτηριστούν κατά περίπτωση βιώσιμες για μία μικρομεσαία επιχείρηση ή οργανισμό είναι άξιοι αναφοράς οι παράγοντες που πρέπει να μελετηθούν και να ληφθούν υπόψιν σχετικά με την επιλογή της καταλληλότερης προσέγγισης. Η φύση των φιλοξενούμενων υπηρεσιών, οι απαιτήσεις τους σε πόρους ή σε ασφάλεια και απομόνωση αποτελούν ένα παράγοντα που θα πρέπει να συνυπολογιστεί σχετικά με το αν η υποδομή περιεκτών πρέπει να είναι ενιαία με την υποδομή εικονικών μηχανών. Οι διαφορετικές υποδομές μεγιστοποιούν την απομόνωση ή απαλλάσσουν από επιπλέον κρίσιμο φόρτο τους φυσικούς πόρους όταν αυτοί είναι περιορισμένοι. Παράλληλα, η δυνατότητα διάθεσης πόρων ή η πολιτική του οργανισμού σχετικά με την αξιοποίηση των πόρων (resource utilization) υποδεικνύουν λύσεις που είτε απαιτούν αύξηση (προσέγγιση 1η) είτε προκαλούν μέγιστη αξιοποίηση (προσέγγιση 2η και 3η) των διαθέσιμων φυσικών πόρων. Τέλος, η πρόβλεψη ή η αδυναμία πρόβλεψης της αύξησης ή μείωσης των υπηρεσιών που φιλοξενούνται θα επηρεάσουν την επιλογή προσέγγισης δεδομένου ότι κάθε προσέγγιση έχει διαφορετικές δυνατότητες επεκτασιμότητας και διαφορετικό επίπεδο δυσκολίας κατά την υλοποίηση της επέκτασης αυτής. Σε κάθε περίπτωση, οι συγκεκριμένες απαιτήσεις της επιχείρησης ή του οργανισμού, το ανθρώπινο δυναμικό που θα υποστηρίξει την λύση, όπως και οι βραχυχρόνιες και μακροχρόνιες προβλέψεις για την φιλοξενία υπηρεσιών θα καθορίσουν ένα συγκεκριμένο πλαίσιο για την μελέτη και επιλογή καταλληλότερης λύσης. Από το πλαίσιο αυτό δεν θα πρέπει να παραλειφθούν και οι πιθανοί μετασχηματισμοί υπηρεσιών που θα προκύψουν είτε

λόγω της τάσης της τεχνολογίας είτε λόγω πολιτικής και θα αναπροσαρμόσουν τις απαιτήσεις των υποδομών φιλοξενίας.

3.2 Μετασχηματισμός εφαρμογών και ενσωμάτωση τους σε περιέκτες

Η τάση που επικρατεί παγκόσμια αναφορικά με τη χρήση της τεχνολογίας των περιεκτών και τη συγγραφή εφαρμογών αρχιτεκτονικής μικροϋπηρεσιών καθορίζει τις ανάγκες για υπολογιστικές υποδομές αλλά παράλληλα συμπαρασύρει και τον μετασχηματισμό εφαρμογών και υπηρεσιών ώστε να ενσωματωθούν σε περιέκτες (containerization) και να μπορούν να εκτελεστούν σε υποδομές φιλοξενίας περιεκτών. Αυτό πολλές φορές συμβαίνει για λόγους ομογενοποίησης και καλύτερης διαχείρισης, αλλά κυρίως για λόγους βελτιστοποίησης της διαχείρισης των πόρων (resource utilization) και ευελιξίας [57], σύμφωνα και με το κυριότερο πλεονέκτημα χρήσης περιεκτών. Από την άλλη πλευρά, συχνά εντοπίζονται επιστημονικές δημοσιεύσεις που επισημαίνουν πως οι προκλήσεις της τεχνολογίας των περιεκτών (υποστήριξη περίπλοκης δικτύωσης, δυνατότητα μόνιμου αποθηκευτικού χώρου, υποστήριξη πολλαπλών κέντρων δεδομένων, ζητήματα ασφαλείας, έλλειψη κατάλληλων προγραμματιστικών διεπαφών κ.λπ.) μειώνουν σημαντικά τον ρυθμό υιοθέτησής της [58].

3.2.1 Παράγοντες και περιορισμοί μετασχηματισμού και ενσωμάτωσης εφαρμογών σε περιέκτες

Η απόφαση ενός πλήρους μετασχηματισμού μιας υποδομής σε υποδομή φιλοξενίας περιεκτών θα οδηγούσε εκ προοιμίου στον πλήρη μετασχηματισμό όλων των υπηρεσιών και εφαρμογών και την ενσωμάτωση τους σε περιέκτες. Η περίπτωση αυτή, ανεξάρτητα από την ορθότητα της, δεν επιτρέπει διερεύνηση ή μεθοδολογική προσέγγιση αναφορικά με την επιλογή εφαρμογών και υπηρεσιών που θα μετασχηματιστούν. Δεδομένου ότι η περίπτωση αυτή δεν κυριαρχεί στο χώρο των μικρομεσαίων επιχειρήσεων και οργανισμών για πληθώρα λόγων σχετικών με το κόστος, την τεχνογνωσία κ.α. είναι άξιοι αναφοράς παράγοντες και κριτήρια που θα πρέπει να ληφθούν υπόψιν πριν την απόφαση μετασχηματισμού.

Οι απαιτήσεις μιας εφαρμογής αποτελούν έναν από τους πρώτους παράγοντες που θα πρέπει να μελετηθούν πριν τη λήψη απόφασης μετασχηματισμού. Το κύριο χαρακτηριστικό των περιεκτών που αφορά την εικονικοποίηση σε επίπεδο

λειτουργικού συστήματος (OS-level virtualization) σε σχέση με τη πλήρη εικονικοποίηση (full virtualization) που προσφέρουν οι εικονικές μηχανές επηρεάζει άμεσα τις δυνατότητες πρόσβασης σε φυσικούς πόρους (π.χ. Peripheral Component Interconnect - PCI cards). Παράλληλα, το γεγονός ότι οι περιέκτες εκτελούνται σε ένα κοινόχρηστο πυρήνα λειτουργικού συστήματος προϋποθέτει την πλήρη **συμβατότητα της εφαρμογής με τον πυρήνα** αυτό.

Ακόμη, οι **απαιτήσεις για ασφάλεια και απομόνωση** μπορούν να αποτελέσουν κρίσιμο παράγοντα απόφασης. Το ανεξάρτητο λειτουργικό σύστημα και πυρήνας σε συνδυασμό με την πλήρη εικονικοποίηση πόρων που προσφέρουν οι εικονικές μηχανές προσφέρουν μεγαλύτερο βαθμό ασφάλειας και απομόνωσης, που ενδέχεται να είναι ζωτικής σημασίας όταν οι εφαρμογές φιλοξενούν ή διαχειρίζονται ευαίσθητες πληροφορίες και δεδομένα.

Επιπρόσθετα, **ιδιαίτερες απαιτήσεις** μιας εφαρμογής οι οποίες δεν είναι συμβατές με την τεχνολογία των περιεκτών δύναται να καθιστούν μη εφικτή ή μη συνιστώμενη αυτήν τη μετατροπή. Ένα συνηθισμένο παράδειγμα αποτελεί η ανάγκη για μόνιμη αποθήκευση μεγάλου όγκου δεδομένων. Το γεγονός ότι οι περιέκτες είναι εφήμεροι χωρίς μόνιμο αποθηκευτικό χώρο (persistent storage) και παράλληλα η δυνατότητα επεκτασιμότητας τους προϋποθέτει ο μόνιμος διασυνδεδεμένος χώρος που χρησιμοποιούν να είναι διαθέσιμος σε οποιοδήποτε κόμβο εκτελείται ο περιέκτης, πρέπει να μελετώνται ή να προβλέπονται προσαρμοσμένες λύσεις αποθήκευσης που στηρίζονται σε κοινόχρηστους χώρους αποθήκευσης.

Επίσης, ζητήματα που σχετίζονται με τον **τρόπο διάθεσης και υποστήριξης** εφαρμογών ή η αδυναμία πρόσβασης στον πηγαίο κώδικα και οι **εξαρτήσεις ή η αδυναμία αποσύνδεσης** (decoupling) από άλλες εφαρμογές ή βιβλιοθήκες αποτελούν τροχοπέδη στο μετασχηματισμό των εφαρμογών αυτές και την ενσωμάτωση τους σε περιέκτες. Πιο αναλυτικά, εάν για την εμπορική υποστήριξη μιας εφαρμογής τίθεται η προϋπόθεση εγκατάστασης της σε συγκεκριμένη υποδομή και με συγκεκριμένη μεθοδολογία τότε ο πιθανός μετασχηματισμός θα έθετε εκτός υποστήριξης την εφαρμογή ανεξάρτητα με τον επιτυχή μετασχηματισμό που μπορεί να επιτυγχανόταν.

Τέλος, η **πολυπλοκότητα της διαδικασίας μετασχηματισμού** ή η **πολυπλοκότητα της μετασχηματισμένης εφαρμογής** αποτελούν κριτήρια προς μελέτη πριν την επιλογή της εφαρμογής για μετατροπή της σε περιέκτες. Η συμβατότητα και η συνάφεια με την αρχιτεκτονική των μικροϋπηρεσιών (microservices) ή την αρχιτεκτονική πολλαπλών επιπέδων (multi-layer applications) που δύναται να μετασχηματιστούν ευκολότερα και να ενσωματωθούν σε περιέκτες ενθαρρύνουν την επιλογή αλλά μετασχηματισμοί που πιθανά δεν μελετώνται επαρκώς δύναται να μειώσουν τις επιδόσεις (π.χ. λόγω διαχείρισης της επικοινωνίας μεταξύ των υποσυστημάτων ή της πρόσβασης σε κοινό αποθηκευτικό χώρο) ή να επιβαρύνουν θέματα σχετικά με την διαχείριση και τη συντήρηση της εφαρμογής.

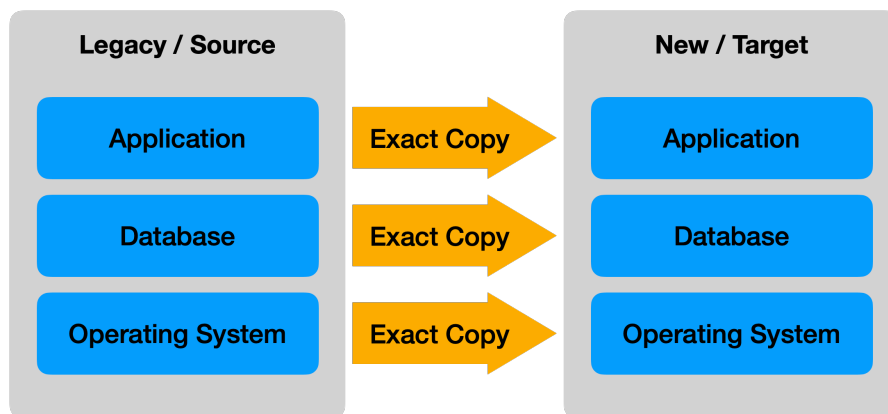
Συνοψίζοντας, ζητήματα ασφαλείας ή συμμόρφωσης με προδιαγραφές, προαπαιτούμενες βιβλιοθήκες και λογισμικά, συμβατότητα του πυρήνα (OS and Kernel Compatibility) και της υποδομής (Storage, Physical Access), απαιτήσεις σχετικές με τον προμηθευτή του λογισμικού (vendor lock-in), όπως και το μέγεθος της προσπάθειας που απαιτείται (effort) αποτελούν τους σημαντικότερους παράγοντες ή πιθανούς περιορισμούς που πρέπει να μελετώνται πριν την εκπόνηση της οποιασδήποτε μεθοδολογίας ή προσέγγισης μετασχηματισμού.

3.2.2 Μεθοδολογίες μετασχηματισμού εικονικών μηχανών και εφαρμογών σε περιέκτες (on-cloud containerization)

Ανεξάρτητα από την αφορμή που έχει οδηγήσει μια επιχείρηση ή έναν οργανισμό στην διαδικασία μετασχηματισμού οι μεθοδολογίες που μπορούν να ακολουθηθούν αποτελούν υποσύνολο των πέντε μεθοδολογιών μετασχηματισμού εφαρμογών και δεδομένων σε υποδομές νέφους που αναγνωρίστηκαν από την Gartner [59] και επεκτάθηκαν μελλοντικά σε έξη 6Rs (Retire, Retain, Repurchase, Rehost, Refactor, Replatform) [60]. Στην περίπτωση των ιδιωτικών νεφούπολογιστικών υποδομών (on-premises cloud infrastructures) και με τις παραδοχές πως ο μετασχηματισμός πραγματοποιείται εντός της ίδιας υποδομής και πιο συγκεκριμένα αφορά εφαρμογές που φιλοξενούνται ήδη σε εικονικές μηχανές οι μεθοδολογίες περιορίζονται κυρίως στις μεθοδολογίες Rehost (Lift-and-Shift), Replatform και Refactor / Rearchitect [61] [62].

3.2.2.1 Rehost (Lift-and-Shift)

Η μεθοδολογία μετασχηματισμού "Lift and Shift" αποτελεί μία από τις πιο διαδεδομένες προσεγγίσεις για τη μετάβαση εφαρμογών από εικονικές μηχανές σε περιέκτες. Η μεθοδολογία αυτή βασίζεται στην απλή, αλλά ολική αντιγραφή του περιεχομένου των εικονικών μηχανών και των εφαρμογών τους, μαζί με το λειτουργικό σύστημα και τις εξαρτήσεις τους, σε έναν περιέκτη.



Σχήμα 7. Μεθοδολογία "Lift & Shift"

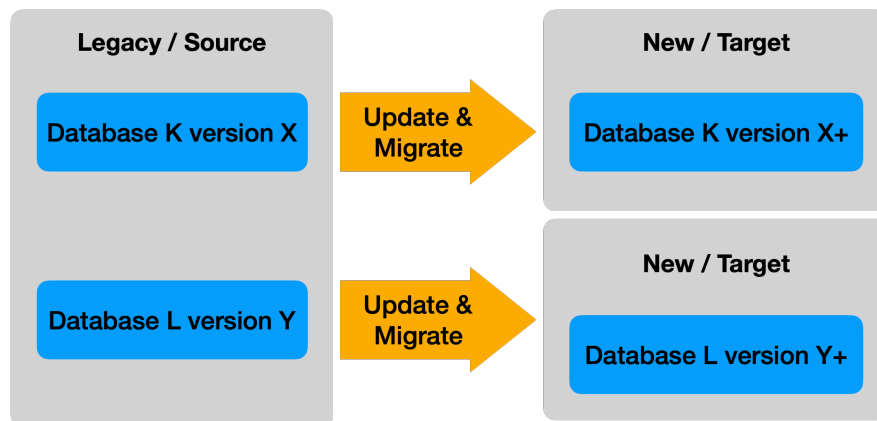
Ο μετασχηματισμός με τη μέθοδο "Lift and Shift" προσφέρει μεγάλο βαθμό ευκολίας και ταχύτητας, αφού δεν απαιτεί μετατροπές ή τροποποιήσεις στον κώδικα των εφαρμογών. Η πρακτική αυτή μειώνει σημαντικά το κόστος και το χρόνο του μετασχηματισμού, ενισχύοντας την επιλογή της όταν οι παράγοντες αυτοί κρίνονται κρίσιμοι. Παράλληλα, η ενσωμάτωση των εξαρτήσεων στο ίδιο περιβάλλον διασφαλίζει την ομαλή μετάβαση ανεξάρτητα πολυπλοκότητας της εφαρμογής .

Ωστόσο, η προσέγγιση αυτή συνήθως δεν αξιοποιεί πλήρως τις δυνατότητες των περιεκτών και δεν εκμεταλλεύεται πλήρως την ευελιξία και την βελτιστοποίηση πόρων που παρέχουν. Συνεπώς, αν και η μέθοδος " Lift and Shift" είναι γρήγορη και εύκολη, δεν μπορεί με βεβαιότητα να οδηγήσει σε βελτίωση της απόδοσης των εφαρμογών σε περιβάλλοντα περιεκτών. Το ενδεχόμενο της μη ωφέλειας των προτερημάτων των περιεκτών και της ανομοιογένειας με τις σύγχρονες αρχιτεκτονικές νέφους ενισχύει την αξιοποίηση της μεθοδολογίας αυτής στις περιπτώσεις που η οποιαδήποτε αναδιαμόρφωση της εικονικής μηχανής ή των εφαρμογών που φιλοξενεί δεν είναι εφικτή ή εύκολη, όπως σε μονολιθικές

εφαρμογές και σε εφαρμογές που ο κώδικας δεν δύναται να τροποποιηθεί λόγω έλλειψης δικαιωμάτων.

3.2.2.2 Replatform ("Lift and reshape", "Lift, Tinker and Shift")

Η μεθοδολογία αυτή αναφέρεται στο μετασχηματισμό της εικονικής μηχανής και της εφαρμογής που φιλοξενεί με τρόπο ώστε ολόκληρη η εφαρμογή ή τμήματα της (application decoupling) να ενσωματώνονται σε διαθέσιμες υπηρεσίες νέφους αξιοποιώντας τις δυνατότητες που παρέχουν. Η διαδικασία συνήθως περιλαμβάνει αναβαθμίσεις, παραμετροποιήσεις και βελτιστοποιήσεις της εφαρμογής. Επιλέγεται όταν αντίστοιχες εφαρμογές είναι διαθέσιμες με τη μορφή υπηρεσιών νέφους. Ένα σύνηθες παράδειγμα αποτελούν οι μετασχηματισμοί εικονικών μηχανών βάσεων δεδομένων με τη μεταφορά τους σε βάσεις δεδομένων ίδιου προμηθευτή αλλά συνήθως νεότερης έκδοσης και πλήρως διαχειρίσιμης από την εκάστοτε νεφοϋπολογιστική πλατφόρμα.

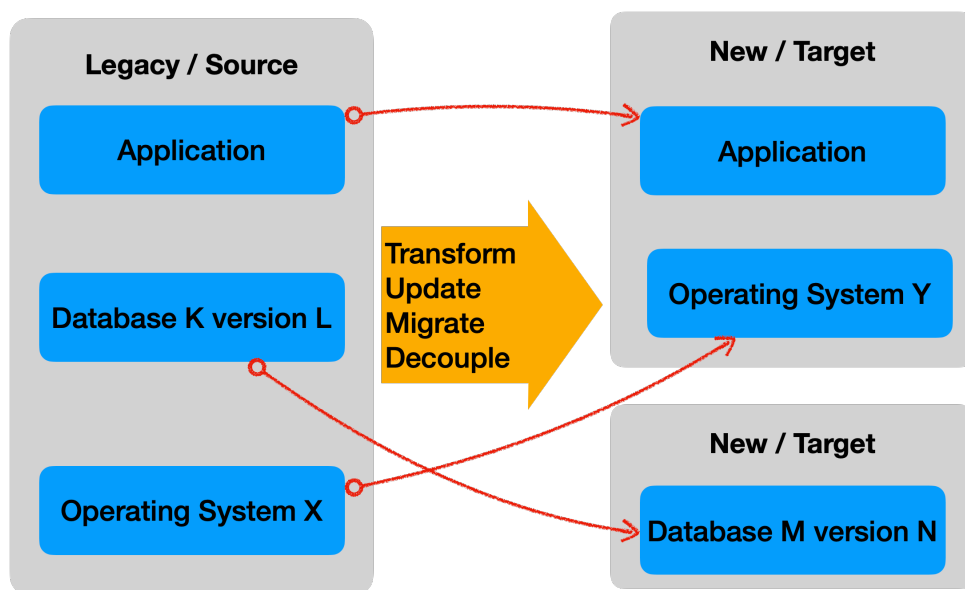


Σχήμα 8. Μεθοδολογία "Replatform"

3.2.2.3 Refactor / Rearchitect

Η τρίτη προσέγγιση αφορά τη μεθοδολογία του "Refactoring" και πρόκειται ίσως για την πιο εξειδικευμένη και περίπλοκη μεθοδολογία αφού εμπεριέχει τον μετασχηματισμό ολόκληρης της εφαρμογής ή τμημάτων της (partial or complete refactor). Συμπληρωματικά από παραμετροποιήσεις και αναβαθμίσεις, μπορεί να περιλαμβάνει τροποποιήσεις στον πηγαίο κώδικα, αλλά και στη δομή / αρχιτεκτονική της εφαρμογής. Ενδείκνυται όταν είναι εφικτός ο διαχωρισμός της

διαστρωμάτωσης των μονολιθικών εφαρμογών (front-end, business logic, backend) ή των οντοτήτων των Service-oriented εφαρμογών τροποποιώντας ή αναβαθμίζοντας τα κατάλληλα υποσυστήματα. Χαρακτηριστικό παράδειγμα της μεθοδολογίας αυτής αποτελεί ο διαχωρισμός μιας εφαρμογής (monolithic, Service-oriented) σε μικρότερες υπηρεσίες αξιοποιώντας ανεξάρτητους περιέκτες και ενσωματώνοντας προγραμματιστικές διεπαφές για την επικοινωνία τους. Η μεθοδολογία αυτή επιβαρύνεται με μεγαλύτερο κόστος και χρόνο αλλά ταυτόχρονα υπόσχεται και τη μεγαλύτερη επιστροφή της επένδυσης (ROI) αφού αξιοποιεί πλήρως τις δυνατότητες των υπηρεσιών νέφους.



Σχήμα 9. Μεθοδολογία "Refactor"

B. ΕΡΕΥΝΗΤΙΚΟ ΜΕΡΟΣ

ΚΕΦΑΛΑΙΟ 4

Μελέτη Περίπτωσης (Case Study)

Η μετάβαση από υποδομές φιλοξενίας εικονικών μηχανών σε υποδομές που φιλοξενούν παράλληλα εικονικές μηχανές και περιέκτες αποτελεί μια πρόκληση για πολλούς οργανισμούς που επιθυμούν να βελτιώσουν την απόδοση και την ευελιξία των εφαρμογών τους, όπως και να ενσωματώσουν ταχύτερα τεχνολογίες αιχμής. Η ανάγκη για πιο αποδοτικές υποδομές και για μείωση των δαπανών συντήρησης και λειτουργίας καθιστά αυτήν τη μετάβαση ουσιώδη. Η παρούσα μελέτη περίπτωσης επικεντρώνεται στη μετάβαση ενός οργανισμού από υποδομές φιλοξενίας εικονικών μηχανών σε υποδομές που φιλοξενούν παράλληλα και εικονικές μηχανές και περιέκτες. Παρουσιάζονται οι ανάγκες και οι προκλήσεις που αντιμετώπισε ο οργανισμός κατά τη μετάβαση αυτή, καθώς και οι λύσεις που υιοθέτησε για να ανταποκριθεί στις ανάγκες του. Μελετάται η απόδοση των συστημάτων και αναλύονται οι δυνατότητες επέκτασης και βελτίωσης τους.

Για τη διεκπεραίωση του παραπάνω προβλήματος, επιλέγεται η μέθοδος της διερευνητικής (exploratory) μελέτης περίπτωσης [63]. Η διερευνητική μελέτη περίπτωσης είναι μια μέθοδος έρευνας που στοχεύει στην διερεύνηση και κατανόηση ενός συγκεκριμένου φαινομένου στο πλαίσιο μιας συγκεκριμένης περίπτωσης. Θεωρείται κατάλληλη μέθοδος όταν δεν υπάρχει επαρκής γνώση για το πρόβλημα και απαιτείται περαιτέρω κατανόηση. Μέσω της διερευνητικής μελέτης περίπτωσης, μπορούν να αναλυθούν λεπτομερώς οι απαιτήσεις του οργανισμού και να εξεταστούν οι διαθέσιμες λύσεις, προτού γίνει μια τελική επιλογή. Με τη χρήση της μεθόδου αυτής, μπορεί να γίνει μια εκτενής ανάλυση των προβλημάτων και των πιθανών λύσεων, καθώς και να αξιολογηθούν οι επιπτώσεις που θα έχει η επιλογή κάποιας λύσης στον οργανισμό. [64]

Η συλλογή δεδομένων εξαρτάται από τους στόχους του οργανισμού και τα κριτήρια που θα χρησιμοποιηθούν για την επιλογή της βέλτιστης υποδομής φιλοξενίας. Κάποια δεδομένα που μπορεί να είναι χρήσιμα για τη λήψη αποφάσεων περιλαμβάνουν:

- Απαιτήσεις υποδομών: Ποιες εφαρμογές θα εκτελούνται στην υποδομή φιλοξενίας και ποιες είναι οι απαιτήσεις τους σε θέματα απόδοσης, διαθεσιμότητας, ασφάλειας και αξιοπιστίας
- Φόρτος εργασίας: Πόσοι χρήστες θα έχουν πρόσβαση στις εφαρμογές και πόσο απαιτητικές θα είναι οι εργασίες τους
- Κόστος: Πόσο θα κοστίσει η υποδομή φιλοξενίας, συμπεριλαμβανομένων των λειτουργικών και μη λειτουργικών απαιτήσεων
- Τεχνογνωσία: Κατά πόσον το προσωπικό έχει τις εξειδικευμένες γνώσεις και τον απαιτούμενο χρόνο για να εγκαταστήσει, διαχειριστεί και συντηρήσει τις υποδομές φιλοξενίας
- Κλιμάκωση: Πώς μπορεί η υποδομή φιλοξενίας να κλιμακωθεί για να αντιμετωπίσει μελλοντικές αυξήσεις στον όγκο της εργασίας;
- Ασφάλεια: Ποια είναι τα απαιτούμενα μέτρα ασφαλείας και προστασίας δεδομένων για τις εφαρμογές

Συνεπώς, η ανάλυση των παραπάνω κριτηρίων είναι ζωτικής σημασίας για την επιλογή της κατάλληλης υποδομής φιλοξενίας εικονικών μηχανών και περιεκτών σε μια μικρής κλίμακας ιδιωτική υποδομή.

4.1 Η περίπτωση του Ελληνικού Κέντρου Θαλάσσιων Ερευνών (ΕΛ.ΚΕ.Θ.Ε.)

Το Ελληνικό Κέντρο Θαλάσσιων Ερευνών (ΕΛ.ΚΕ.Θ.Ε.) είναι κρατικός ερευνητικός οργανισμός που λειτουργεί υπό την εποπτεία της Γενικής Γραμματείας Έρευνας και Καινοτομίας (ΓΓΕΚ) του Υπουργείου Ανάπτυξης και Επενδύσεων. Το ΕΛ.ΚΕ.Θ.Ε. αποτελείται από τρία ερευνητικά Ινστιτούτα: το Ινστιτούτο Θαλάσσιας Βιολογίας, Βιοτεχνολογίας και Υδατοκαλλιεργειών, Ινστιτούτο Θαλάσσιων Βιολογικών Πόρων και Εσωτερικών Υδάτων και το Ινστιτούτο Ωκεανογραφίας.

Το Ελληνικό Κέντρο Θαλασσίων Ερευνών ασχολείται με τη μελέτη, την προστασία και τη βιώσιμη διαχείριση των θαλασσίων οικοσυστημάτων στην Ελλάδα. Το κέντρο ιδρύθηκε το 1987 και αποτελεί έναν από τους σημαντικότερους φορείς θαλάσσιας έρευνας στη χώρα.

Η αποστολή του ΕΛΚΕΘΕ περιλαμβάνει την προώθηση της επιστημονικής έρευνας σε θέματα που αφορούν το θαλάσσιο περιβάλλον, την αειφόρο αξιοποίηση των θαλάσσιων πόρων, την προστασία της θαλάσσιας βιοποικιλότητας και την παροχή επιστημονικής υποστήριξης στην πολιτική λήψη αποφάσεων και την ανάπτυξη της θαλάσσιας πολιτικής.

Οι ερευνητικές δραστηριότητες του ΕΛΚΕΘΕ περιλαμβάνουν τη μελέτη των ωκεανών, την έρευνα πάνω σε θαλάσσια είδη και οικοσυστήματα, την παρακολούθηση της ποιότητας του θαλάσσιου περιβάλλοντος, την ανάπτυξη εργαλείων εκτίμησης της ποιότητας των ποταμιών και των υδροτοπικών οικοσυστημάτων, την ανάπτυξη μεθοδολογιών οικολογικής αξιολόγησης, όπως και ορθών πρακτικών περιβαλλοντικής διαχείρισης και αποκατάστασης και συμβάλλει καίρια στην αλιευτική έρευνα. Επίσης εστιάζει στις υδατοκαλλιέργειες, τη γονιδιωματική και τη μοριακή βιοτεχνολογία, τη θαλάσσια βιοποικιλότητα και τη διαχείριση των οικοσυστημάτων.

Το ΕΛΚΕΘΕ συνεργάζεται με διάφορους εθνικούς και διεθνείς φορείς, περιλαμβανομένων πανεπιστημίων, ερευνητικών κέντρων, κυβερνητικών φορέων και μη κυβερνητικών οργανώσεων, με στόχο την ανταλλαγή γνώσεων, τη συνεργασία σε έρευνες και την προώθηση της βιώσιμης διαχείρισης των θαλάσσιων πόρων. Σε αυτά τα πλαίσια λειτουργούν επιχειρησιακά το Σύστημα παρακολούθησης και πρόβλεψης θαλάσσιου περιβάλλοντος Ποσειδών όπως και το Σύστημα παρακολούθησης εσωτερικών υδάτων για τη διαχείριση ανθρωπογενών και φυσικών καταστροφών.

Το ΕΛΚΕΘΕ διαθέτει τρία ερευνητικά σκάφη, ένα υποβρύχιο σκάφος, θαλάσσια παρατηρητήρια και πολλαπλά όργανα πεδίου. Επίσης έχει δύο κύριες εγκαταστάσεις στο Ηράκλειο Κρήτης και στην Ανάβυσσο Αττικής, δύο δευτερεύουσες στην Αργυρούπολη Αττικής και στη Ρόδο και διάφορα σημεία παρουσίας ανά την Ελλάδα (π.χ. Καλαμάτα, Χανιά κλπ.), ενώ διαθέτει και δύο ενυδρεία διεθνών προδιαγραφών που στοχεύουν να τονώσουν το ενδιαφέρον, να αναπτύξουν και να διαδώσουν τη γνώση σχετικά με το θαλάσσιο περιβάλλον εστιάζοντας κυρίως στην Ανατολική περιοχή της Μεσογείου.

Στα πλαίσια υποστήριξης των παραπάνω, το ΕΛΚΕΘΕ διαθέτει τις δικές του υπολογιστικές υποδομές οι οποίες φιλοξενούν τις ετερόκλητες ανάγκες των τριών

Ινστιτούτων αλλά και των διοικητικών υπηρεσιών. Οι υπολογιστικές υποδομές που θα συναντήσει κάποιος στις εγκαταστάσεις του ΕΛΚΕΘΕ μπορούν να χαρακτηριστούν ιδιαίτερα σύγχρονες και φιλοξενούνται σε τρία Κέντρα Δεδομένων (Data Centers - DC) τα οποία απαρτίζονται από δικτυακές, υπολογιστικές και αποθηκευτικές υποδομές, οι οποίες υλοποιούνται σε συστοιχίες και υλοποιούν τεχνολογίες υψηλής διαθεσιμότητας και ανοχής σφαλμάτων.

Πιο συγκεκριμένα, τα τρία DC διαθέτουν συνδεσιμότητα με το διαδίκτυο μέσω οπτικών ινών αξιοποιώντας το Εθνικό Δίκτυο Υποδομών Τεχνολογίας και Έρευνας, ενώ παράλληλα διαθέτουν και υποστηρικτική συνδεσιμότητα χαμηλότερης ταχύτητας μέσω τρίτων παροχών. Όλο το δίκτυο κορμού στηρίζεται σε συνδέσεις υψηλής ταχύτητας της τάξης των 40Gbps και προστατεύεται από Τείχη Προστασίας (Firewalls) και Συστήματα Ανίχνευσης Εισβολών (Intrusion Detection Systems - IDS). Για την κάλυψη υψηλών υπολογιστικών αναγκών φιλοξενούνται τέσσερα Υπολογιστικά Συστήματα Υψηλής Απόδοσης (High Performance Computing - HPC) τα οποία εκτελούν, σε επιχειρησιακή ή ερευνητική βάση, μετεωρολογικά ή γενετικά μοντέλα. Παράλληλα για την υποστήριξη των υπόλοιπων ερευνητικών και διοικητικών αναγκών έχουν υλοποιηθεί νεφούπολογιστικές υποδομές εικονικών μηχανών με δυνατότητες υψηλής διαθεσιμότητας και καταμερισμού φόρτου. Όλα τα υπολογιστικά συστήματα διατηρούν συνδέσεις με συστήματα αποθήκευσης δεδομένων που αξιοποιούν τεχνολογίες δευτερεύουσας αποθήκευσης (RAID), διαθέτουν πολλαπλά επίπεδα ταχύτητας και χωρητικότητας, ενώ παράλληλα λαμβάνονται αντίγραφα ασφαλείας σε ξεχωριστή γεωχωρική τοποθεσία προσομοιάζοντας λύσεις Αποκατάστασης Καταστροφών (Disaster Recovery - DR). Τέλος, για την παρακολούθηση της ορθής λειτουργίας των υπολογιστικών και δικτυακών συστημάτων έχει εγκατασταθεί πλήρες σύστημα παρακολούθησης σε πραγματικό χρόνο.

4.2 Περίληψη – Περιγραφή προβλήματος

Οι ερευνητικοί οργανισμοί αντιμετωπίζουν συνεχώς την ανάγκη για προηγμένες υποδομές που να υποστηρίζουν τις απαιτήσεις τους στον τομέα της επιστημονικής έρευνας και της καινοτομίας. Ενώ οι υπάρχουσες υποδομές εικονικών μηχανών παρέχουν ευελιξία και πληθώρα δυνατοτήτων, η ανάπτυξη υποδομών που μπορούν να φιλοξενήσουν περιέκτες αποτελεί μια αναδυόμενη ανάγκη. Οι

περιέκτες προσφέρουν ακόμα μεγαλύτερη ευελιξία, φορητότητα και αποτελεσματικότητα στην ανάπτυξη και στην ενσωμάτωση εφαρμογών. Επιπρόσθετα η χρήση περιεκτών έχει γίνει ένα από τα πιο διαδεδομένα μοντέλα διανομής εφαρμογών.

Η αύξηση της χρήσης των περιεκτών στην ανάπτυξη και διανομή εφαρμογών οδηγεί σε αυξημένη ζήτηση για υποδομές φιλοξενίας περιεκτών. Οι οργανισμοί βλέπουν την αξία των περιεκτών στην ευελιξία, την αποτελεσματικότητα, την παραγωγικότητα και την ευκολία διαχείρισης που προσφέρουν. Σε συνδυασμό με την ανάγκη για ελαστικότητα και κλιμάκωση στις σύγχρονες εφαρμογές, καθιστά αναγκαία την υποστήριξη της τεχνολογίας των περιεκτών στις ερευνητικές και επαγγελματικές υποδομές. Έτσι, η δημιουργία υποδομών που μπορούν να φιλοξενήσουν περιέκτες γίνεται ολοένα και πιο σημαντική για τους οργανισμούς που επιθυμούν να παραμείνουν ανταγωνιστικοί στον ταχέως εξελισσόμενο κόσμο της τεχνολογίας.

Αντίστοιχα, στο ΕΛΚΕΘΕ η ζήτηση για υποδομές που μπορούν να φιλοξενήσουν περιέκτες γίνεται όλο και πιο έντονη, είτε μέσα από ομάδες έρευνας και ανάπτυξης, είτε μέσα από προγράμματα, είτε από συνεργαζόμενους φορείς που διανέμουν το λογισμικό που παράγουν σε μορφή περιεκτών. Επιπλέον η διαδεδομένη προσέγγιση αρχιτεκτονικής ανάπτυξης εφαρμογών με μικροϋπηρεσίες, δηλαδή ο διαχωρισμός μιας εφαρμογής σε πολλές μικρές ανεξάρτητες υπηρεσίες που επικοινωνούν μεταξύ τους, η οποία υλοποιείται με την χρήση περιεκτών, ενδυναμώνει σταθερά την ζήτηση σε αντίστοιχες υποδομές φιλοξενίας.

Για την αντιμετώπιση της προκύπτουσας ανάγκης καλείται η ομάδα διαχείρισης συστημάτων και δικτύων του Φορέα να μελετήσει και να υλοποιήσει πιθανές λύσεις ώστε προγραμματιστές, ερευνητικές ομάδες και εξωτερικοί συνεργάτες να μπορούν να εγκαταστήσουν και να διαχειριστούν εφαρμογές που βασίζονται σε μικροϋπηρεσίες και περιέκτες.

Στην παρούσα μελέτη αναφέρονται οι διάφορες λύσεις που μελετήθηκαν και προτάθηκαν προς δοκιμή από τους διαχειριστές δικτύων και συστημάτων του Φορέα. Οι λύσεις αυτές υλοποιήθηκαν βάσει των διαθέσιμων υποδομών και της υφιστάμενης τεχνογνωσίας.

4.3 Μελέτη υπάρχουσας κατάστασης και Εκτίμηση λύσεων

Για την αντιμετώπιση του προβλήματος που αναγνωρίστηκε και την εξεύρεση κατάλληλων λύσεων, οι διαχειριστές συστημάτων και δικτύων του ΕΛΚΕΘΕ, αναζήτησαν κατάλληλες μεθοδολογίες και διαδικασίες, ώστε να μπορέσουν να εκτιμήσουν ολιστικά το πρόβλημα και να αναζητήσουν λύσεις οι οποίες θα βασίζονται σε καλές πρακτικές και θα είναι επιστημονικά τεκμηριωμένες.

Η προσέγγιση που ακολουθήθηκε αποτελεί μια γενική πρακτική που περιλαμβάνει διάφορες μεθόδους και τεχνικές για την επιλογή τεχνολογικών λύσεων. Γενικότερα, μπορεί να θεωρηθεί ως μέρος της διαδικασίας της "Διαχείρισης Πληροφοριακών Συστημάτων" (Information Systems Management) ή της "Διαχείρισης Τεχνολογίας Πληροφορικής" (Information Technology Management). [65] [66]

Τα στάδια που απαρτίζουν την γενική αυτή προσέγγιση περιγράφονται ακολούθως

- **Ανάλυση Απαιτήσεων:** Το πρώτο βήμα είναι η κατανόηση των απαιτήσεων του φορέα. Αυτό μπορεί να περιλαμβάνει την αναγνώριση των λειτουργικών αναγκών, των τεχνολογικών περιορισμών και των προτιμήσεων και δυνατοτήτων των χρηστών.
- **Βιβλιογραφική Ανασκόπηση:** Η ανασκόπηση της βιβλιογραφίας παρέχει πολύτιμες πληροφορίες σχετικά με τη χρήση, την αποτελεσματικότητα και τις πιθανές προκλήσεις των διαθέσιμων τεχνολογικών λύσεων. Διερευνά τα πρόσφατα ευρήματα της έρευνας, τις απόψεις των ειδικών, τις περιπτώσεις χρήσης και τις εμπειρίες άλλων οργανισμών. Αυτό μπορεί να βοηθήσει στην κατανόηση του πώς μπορεί να εφαρμοστεί μια τεχνολογία στα πλαίσια του φορέα και ποιες μπορεί να είναι οι επιπτώσεις.
- **Τεχνική Αξιολόγηση:** Καταγράφονται οι διάφορες τεχνολογικές λύσεις που είναι διαθέσιμες και αξιολογούνται από τεχνικής πλευράς. Αυτό περιλαμβάνει την εξέταση των δυνατοτήτων, της απόδοσης, της ασφάλειας, της ευελιξίας και της αξιοπιστίας των λύσεων.
- **Οικονομική Ανάλυση:** Η οικονομική ανάλυση είναι επίσης σημαντικό στοιχείο της διαδικασίας επιλογής τεχνολογικών λύσεων. Αυτό περιλαμβάνει την κατανόηση του κόστους απόκτησης, εφαρμογής και

συντήρησης της τεχνολογίας, καθώς και της απόδοσης της επένδυσης (Return Of Investment - ROI).

- **Απόφαση:** Σε αυτό το στάδιο, αφού έχουν ληφθεί υπόψη όλα τα παραπάνω στοιχεία - αναλύσεις, βιβλιογραφική ανασκόπηση, σύγκριση λύσεων, είναι η στιγμή όπου οι εμπλεκόμενοι παίρνουν την τελική απόφαση. Η απόφαση αυτή πρέπει να λαμβάνει υπόψη την τεχνογνωσία και τις δυνατότητες του ανθρώπινου δυναμικού που θα χρησιμοποιήσει και θα συντηρήσει την τεχνολογία.
- **Εφαρμογή:** Μετά την απόφαση, η επιλεγμένη τεχνολογία πρέπει να εγκατασταθεί και να ενσωματωθεί στον οργανισμό. Αυτό περιλαμβάνει την εκπαίδευση του προσωπικού, την προσαρμογή των εργασιακών διαδικασιών και, ενδεχομένως, την αλλαγή της οργανωτικής κουλτούρας. Οι ανάγκες του ανθρώπινου δυναμικού πρέπει να λαμβάνονται σοβαρά υπόψη κατά την εφαρμογή. Αυτό μπορεί να περιλαμβάνει την παροχή κατάρτισης, την αλλαγή των οργανωτικών δομών ή την προσαρμογή των ρόλων και των ευθυνών για να εξασφαλίσει την αποτελεσματική χρήση και συντήρηση της νέας τεχνολογίας. Επίσης, αυτό το στάδιο πρέπει να περιλαμβάνει επαρκή χρόνο για την εξοικείωση και την προσαρμογή στη νέα τεχνολογία. Αυτό είναι κρίσιμο, καθώς η επιτυχής ενσωμάτωση μιας νέας τεχνολογίας δεν εξαρτάται μόνο από τις τεχνικές δυνατότητες της, αλλά και από την ικανότητα των ανθρώπων να την χρησιμοποιήσουν και να την διαχειριστούν αποτελεσματικά.

4.3.1 Ανάλυση απαιτήσεων

4.3.1.1 Ανάλυση Λειτουργικών Απαιτήσεων ΕΛΚΕΘΕ

Η πρώτη και κύρια λειτουργική απαίτηση που προέκυψε από πολύ νωρίς είναι η δυνατότητα εκτέλεσης αυτόνομων και ανεξάρτητων περιεκτών, οι οποίοι συνήθως ενσωμάτωναν την εφαρμογή που είχε αναπτύξει κάποιος προγραμματιστής ή ερευνητική ομάδα. Αυτό οφειλόταν στην διαδεδομένη πλέον πρακτική, η διανομή εφαρμογών να υλοποιείται μέσω περιεκτών ώστε να επιτυγχάνεται η εύκολη φορητότητα και αναβάθμιση της εφαρμογής.

Η δεύτερη απαίτηση, η οποία είναι συμπληρωματική της πρώτης, σχετίζεται με την ανάγκη εγκατάστασης και διαχείρισης εφαρμογών που έχουν αναπτυχθεί με βάση την αρχιτεκτονική των μικροϋπηρεσιών. Η απαίτηση αυτή προήλθε κυρίως από την ανάγκη εγκατάστασης εφαρμογών ανοικτού κώδικα, οι οποίες πλέον εγκαταλείπουν την μονολιθική αρχιτεκτονική και προσφέρονται κατά βάσει με την αρχιτεκτονική μικροϋπηρεσιών, αξιοποιώντας παράλληλα και την τεχνολογία των APIs.

Η απαίτηση για εύκολη και γρήγορη διανομή ταυτόσημων υπηρεσιών, όπως οι βάσεις δεδομένων (SaaS) ή ο συνδυασμός βάσεων δεδομένων με διαχειριστικό περιβάλλον (π.χ. Postgres σε συνδυασμό με pgAdmin ή MySQL σε συνδυασμό με phpMyAdmin - PaaS), καλύπτεται άριστα από τους περιέκτες. Οι περιέκτες προσφέρουν εξαιρετικές δυνατότητες διανομής ίδιων υπηρεσιών, καθώς μπορεί εύκολα να δημιουργηθεί και να διανεμηθεί μια εικόνα περιέκτη, η οποία θα ενσωματώνει το απαραίτητο λογισμικό και θα μπορεί να επαναχρησιμοποιηθεί σε οποιοδήποτε περιβάλλον υποστηρίζει περιέκτες.

Οι ερευνητικές ομάδες που δραστηριοποιούνται στον Φορέα και αναπτύσσουν λογισμικό ακολουθούν πρακτικές της συνεχούς ενοποίησης, παράδοσης και ανάπτυξης (CI/CD), οι οποίες βασίζονται κατά κύριο λόγο σε εφαρμογές αρχιτεκτονικής μικροϋπηρεσιών και περιεκτών. Η υποστήριξη αυτού του συνόλου πρακτικών απαιτεί αντίστοιχη υποδομή διαχείρισης και ενορχήστρωσης περιεκτών.

Η ύπαρξη εφαρμογών που ο φόρτος τους είναι κυμαινόμενος ανάλογα με τις περιστάσεις οδηγεί στον μετασχηματισμό τους σε εφαρμογές αρχιτεκτονικής μικροϋπηρεσιών ώστε να αξιοποιείται κατά το μέγιστο η ευελιξία στην κλιμάκωση που προσφέρεται από την συγκεκριμένη αρχιτεκτονική. Χαρακτηριστικά παραδείγματα που αντικατοπτρίζουν την ανάγκη αυτή αποτελούν το Σύστημα πρόγνωσης καιρού «Ποσειδών» και τα συστήματα μετρητικών σταθμών εσωτερικών υδάτων για φυσικές καταστροφές, τα οποία δέχονται τρομακτική αύξηση των αιτημάτων σε περιόδους που αναμένονται ή υφίστανται ακραία καιρικά φαινόμενα και τα οποία προφανώς δεν μπορούν να προβλεφθούν ώστε να διαμορφωθούν κατάλληλα οι πόροι των υποδομών που τα φιλοξενούν.

Τέλος, θα πρέπει να διατηρηθεί η δυνατότητα του Φορέα να μπορεί να προσφέρει νεφούπολογιστικές υπηρεσίες εικονικών μηχανών (IaaS) παράλληλα ή ανεξάρτητα

από τις υπηρεσίες φιλοξενίας περιεκτών, δεδομένου ότι μεγάλο πλήθος εφαρμογών και υπηρεσιών συνεχίζουν να φιλοξενούνται σε εικονικές μηχανές. Αυτό συμβαίνει λόγω της αρχιτεκτονικής των εφαρμογών αυτών, της δυσκολίας μετασχηματισμού τους ή ακόμα και της μη ύπαρξης των απαραίτητων πόρων (π.χ. ανθρωποώρες, έλλειψης τεχνογνωσίας κλπ.) για τον μετασχηματισμό αυτό.

4.3.1.2 Ανάλυση Τεχνολογικών Περιορισμών ΕΛΚΕΘΕ

Παρόλα τα πλεονεκτήματα που μπορεί να προσφέρει σε έναν Φορέα η τεχνολογία των περιεκτών σε συνδυασμό με τις υποδομές φιλοξενίας τους, υπάρχουν κάποιοι τεχνολογικοί περιορισμοί που πρέπει να ληφθούν υπόψιν και μπορεί να επηρεάσουν την εγκατάσταση υποδομών εκτέλεσης και ενορχήστρωσης περιεκτών. Παρακάτω αναλύονται συνοπτικά οι κυριότεροι εξ' αυτών.

Συμβατότητα Συστήματος: Οι περιέκτες απαιτούν λειτουργικό σύστημα που υποστηρίζει containerization, όπως το Linux. Εάν ένας οργανισμός χρησιμοποιεί ένα λειτουργικό σύστημα που δεν υποστηρίζει τους περιέκτες, θα χρειαστεί να κάνει ενημερώσεις ή αλλαγές πριν να μπορέσει να χρησιμοποιήσει περιέκτες. Το ΕΛΚΕΘΕ χρησιμοποιεί κατά κόρον περιβάλλοντα Linux, κάτι το οποίο όχι μόνο αποσύρει τον συγκεκριμένο περιορισμό, αλλά ενθαρρύνει την υλοποίηση τέτοιων τεχνολογιών. Αφού η πληθώρα υποδομών ενορχήστρωσης υλοποιείται μέσα από συστήματα Linux.

Πόροι Υλικού: Η εκτέλεση περιεκτών και οι αντίστοιχες υποδομές ενορχήστρωσης και εκτέλεσης περιεκτών απαιτούν σημαντικούς πόρους υλικού, συμπεριλαμβανομένης της μνήμης, της επεξεργαστικής ισχύος και του αποθηκευτικού χώρου. Αυτό σημαίνει ότι θα πρέπει να καθοριστούν με σαφήνεια και ακρίβεια οι ανάγκες για υπολογιστικούς πόρους ώστε είτε να επιλεχθούν οι κατάλληλες υποδομές για περιέκτες που να καλύπτονται από τους υπάρχοντες πόρους, είτε να γίνουν οι απαραίτητες αναβαθμίσεις ώστε να ικανοποιούνται οι απαιτήσεις των αντίστοιχων προδιαγραφών που ορίζονται από τις υποδομές ενορχήστρωσης.

Διαχείριση Περιεκτών: Το να διαχειρίζεσαι μεμονωμένους περιέκτες μπορεί να είναι σχετικά εύκολο, αλλά όταν οι περιέκτες πολλαπλασιάζονται και πρέπει να λειτουργήσουν συντονισμένα, τα πράγματα γίνονται πιο πολύπλοκα. Αυτή η πολυπλοκότητα επιβάλλει τη χρήση εργαλείων διαχείρισης περιεκτών όπως το

Kubernetes, το Docker Swarm ή το OpenShift, τα οποία αυτοματοποιούν την εγκατάσταση, την κλιμάκωση, την κατανομή δικτύου και τη διαχείριση της ζωής των περιεκτών. Επιπλέον, είναι απαραίτητο να υπάρχει επαρκής παρακολούθηση και αποσφαλμάτωση (debugging) των περιεκτών. Εργαλεία όπως το Prometheus ή το Grafana μπορούν να βοηθήσουν σε αυτό, παρέχοντας δυνατότητες παρακολούθησης και απεικόνισης (visualization) για τις εφαρμογές περιεκτών. Σημαντική πτυχή της διαχείρισης περιεκτών είναι επίσης η διαχείριση των εικόνων περιεκτών. Πρέπει να υπάρχουν διαδικασίες για τη δημιουργία, την αποθήκευση, την ενημέρωση και την ανάλυση των εικόνων περιεκτών, ώστε να διασφαλίζεται η ποιότητα και η ασφάλεια των εφαρμογών που εκτελούνται μέσα στους περιέκτες. Όλα αυτά απαιτούν εξειδικευμένες γνώσεις και χρόνο για την εκμάθηση και την εφαρμογή τους, πράγμα που πρέπει να ληφθεί υπόψη κατά την απόφαση για την υιοθέτηση τεχνολογίας περιεκτών. Στο ΕΛΚΕΘΕ δεν υπάρχει προϋπάρχουσα τεχνογνωσία σε τεχνολογίες ενορχήστρωσης περιεκτών και αυτό καθιστά ακόμα πιο δαιδαλώδη την απόφαση για την επιλογή μιας λύσης της οποίας η καμπύλη εκμάθησης (learning curve) θα είναι όσο το δυνατόν μικρότερη. Αυτό θα διασφαλίσει την απρόσκοπτη λειτουργία των υπόλοιπων υπηρεσιών αλλά και οι εμπλεκόμενοι να μπορούν γρήγορα να διαχειριστούν (διαχειριστές) ή να μεταβούν (προγραμματιστές) παραγωγικά στο νέο περιβάλλον.

Ασφάλεια: Η ασφάλεια των περιεκτών αποτελεί κρίσιμο παράγοντα και απαιτεί προσεκτική μελέτη. Οι περιέκτες μοιράζονται τον πυρήνα του συστήματος που τους φιλοξενεί, κάτι που θα μπορούσε ενδεχομένως να τους εκθέσει σε κινδύνους ασφαλείας. Η ασφάλεια των εικόνων των περιεκτών είναι επίσης σημαντική, καθώς πρέπει να είναι ελεγμένες για ευπάθειες και να προέρχονται από αξιόπιστες πηγές, όπως αποθετήρια εικόνων τύπου Docker Hub ή ακόμα και από τους επίσημους ιστοτόπους των επιθυμητών εφαρμογών. Η βέλτιστη και πιο αξιόπιστη λύση είναι η χρήση ιδιωτικών αποθετηρίων εικόνων, όπου υπάρχουν εικόνες περιεκτών ελεγμένες από τον ίδιο τον οργανισμό, ο οποίος θέλει να κάνει χρήση. Επιπλέον, η χρήση ελάχιστων δικαιωμάτων (principle of least privilege) και η διαχείριση των ευπαθειών είναι ζωτικής σημασίας για την διασφάλιση της ασφαλείας των περιεκτών [67]. Στο ΕΛΚΕΘΕ λόγω της περιορισμένης τεχνογνωσίας μελετάται ως πιθανή η χρήση εικόνων μόνο από επίσημους φορείς ή εικόνες που έχουν αναπτυχθεί από ομάδες εντός του Φορέα και οι οποίες θα δημιουργήσουν εν

συνεχία το ιδιωτικό αποθετήριο εικόνων περιεκτών (private container image registry).

4.3.1.3 Ανάλυση των προτιμήσεων και δυνατοτήτων των χρηστών ΕΛΚΕΘΕ

Η ανάλυση των προτιμήσεων και των δυνατοτήτων των χρηστών είναι μια σημαντική διαδικασία κατά την μετάβαση σε νέα τεχνολογία. Αυτό επιτυγχάνεται μέσω της συλλογής και της ανάλυσης πληροφοριών για τους χρήστες και την προσαρμογή της τεχνολογίας στις ανάγκες τους. Η συλλογή και η ανάλυση των πληροφοριών οργανώθηκε σε τρεις βασικές θεματικές ομάδες οι οποίες είναι η κατανόηση των χρηστών, η αξιολόγηση της νέας τεχνολογίας ως προς την φιλικότητα της προς τον χρήστη και τέλος η απαιτούμενη εκπαίδευση και υποστήριξη των χρηστών. Παρακάτω αναλύεται ξεχωριστά η κάθε θεματική ομάδα.

Κατανόηση των χρηστών: Η θεματική ομάδα αυτή, αναφέρεται στην αναγνώριση της εξοικείωσης των χρηστών με σχετικές τεχνολογίες και στην κατανόηση των δυνατοτήτων τους. Το ΕΛΚΕΘΕ περιλαμβάνει τρεις κύριες κατηγορίες χρηστών, ερευνητικές ομάδες που παράγουν λογισμικό, εξωτερικούς συνεργάτες που συνήθως αξιοποιούν τεχνολογίες περιεκτών Docker με λογισμικό ανοιχτού κώδικα, διαδεδομένες στην επιστημονική κοινότητα (π.χ. ERDDAP, ThingsBoard) και το προσωπικό της διαχείρισης συστημάτων και δικτύων του Φορέα. Οι απαιτούμενες πληροφορίες αντλήθηκαν κυρίως μέσα από συνεντεύξεις οι οποίες είχαν διερευνητικό χαρακτήρα, εκπονήθηκαν στα πλαίσια άτυπων συναντήσεων και ανάλογα με τον κάθε χρήστη υπήρξε δυναμική αναπροσαρμογή της ροής στην συνάντηση. Για παράδειγμα με χρήστες που ήδη είχαν επαφή με τεχνολογίες περιεκτών συζητήθηκαν θέματα σχετικά με την αυτοματοποίηση, την κατανομή πόρων και τις τεχνικές συνεχούς ενσωμάτωσης και διανομής (CI/CD), ενώ με χρήστες που δεν είχαν γνώση σχετικών τεχνολογιών, έγινε μια εισαγωγή στην τεχνολογία των περιεκτών και διερευνήθηκε η διάθεση για ενασχόληση και ο τρόπος διανομής των υπηρεσιών αυτών που θα προσέφερε την ομαλότερη εμπειρία χρήσης. Από τις συνεντεύξεις αυτές προέκυψε με σαφήνεια ότι οι περισσότεροι εμπλεκόμενοι έχουν εξοικείωση με σχετικές τεχνολογίες - τουλάχιστον σε θεωρητικό επίπεδο - όπως και την πρόθεση για περαιτέρω ενασχόληση. Από την άλλη πλευρά η διαχείριση των υποδομών κρίνεται βέλτιστο να πραγματοποιείται από κατάλληλα καταρτισμένο προσωπικό (ομάδα διαχείρισης

συστημάτων και δικτύων) και όχι από τους ίδιους τους χρήστες. Η ομάδα διαχείρισης συστημάτων και δικτύων διαθέτει επαρκές υπόβαθρο για να ασχοληθεί με την ανάπτυξη και την διαχείριση τέτοιων υποδομών, αλλά δεν διαθέτει εξειδίκευση και εμπειρία. Όλα τα παραπάνω προάγουν λύσεις που θα έχουν δυνατότητες αυτοματοποίησης, δικλίδες ασφαλείας και γραφικό περιβάλλον χρήστη.

Αξιολόγηση της νέας τεχνολογίας: Είναι σημαντικό να αξιολογηθεί πόσο εύκολη είναι η χρήση της νέας τεχνολογίας για τους χρήστες και πόσο εύκολα μπορούν να εξοικειωθούν με αυτήν. Κατά κύριο λόγο, αυτού του είδους οι υποδομές απαιτούν χρήστες που επιλέγουν συνειδητά την αξιοποίηση τους και διαθέτουν επαρκή αντίληψη των τρόπων διανομής νεφούπολογιστικών υποδομών και της διαρκούς ανάγκης για εκπαίδευση. Η παρατήρηση αυτή διευρύνει την επιλογή λύσεων σε σχέση με το διαχειριστικό περιβάλλον, διατηρώντας σαφώς την αναγκαιότητα ύπαρξης γραφικού περιβάλλοντος, όπως διαπιστώθηκε και στην προηγούμενη θεματική ενότητα. Παράλληλα η πολυπλοκότητα διαχείρισης τέτοιων υποδομών, η διασφάλιση ορθής κατανομής πόρων και η ανάγκη για περίπλοκη παραμετροποίηση συνιστούν ένα ιδιαίτερα απαιτητικό οικοσύστημα τεχνολογιών. Αυτό μας οδηγεί στην επιλογή λύσης που θα εμπεριέχει ικανοποιητική τεκμηρίωση, αυτοματοποιημένες διαδικασίες διαχείρισης (εγκατάσταση, επέκταση, αναβάθμιση κλπ.) και εμπορική υποστήριξη κατά απαίτηση.

Εκπαίδευση και υποστήριξη: Στις περισσότερες περιπτώσεις υιοθέτησης νέων τεχνολογιών συνίσταται η εκπαίδευση και η υποστήριξη των χρηστών ώστε να ενθαρρυνθεί η χρήση τους, να αξιοποιηθούν στο μέγιστο βαθμό και τελικά να αφομοιωθούν στον φορέα. Στην παρούσα περίπτωση και δεδομένου του περιορισμένου πλήθους χρηστών προτάθηκε και προκρίθηκε η εκπαίδευση των χρηστών να πραγματοποιηθεί στα πλαίσια ευέλικτων ομάδων εκπονώντας περιπτώσεις χρήσης (use cases) των υποδομών αυτών, αφού πρώτα η ομάδα διαχείρισης συστημάτων και δικτύων θα έχει εντρυφήσει σε μεγαλύτερο βάθος και θα έχει δημιουργήσει σενάρια χρήσης των προσφερόμενων υπηρεσιών.

Το ΕΛΚΕΘΕ διαθέτει IaaS και δύναται να υλοποιήσει περιέκτες σε εικονικές μηχανές. Αυτή η λύση μπορεί να θεωρηθεί αρκετά αποτελεσματική και πρόκειται για έναν σχετικά απλά τρόπο για τη φιλοξενία περιεκτών στις υποδομές του. Παρόλα αυτά η συγκεκριμένη λύση δεν αξιοποιεί πλήρως τις δυνατότητες των

περιεκτών και θα μπορούσε να χαρακτηριστεί αρκετά σπάταλη σε υπολογιστικούς πόρους, γεγονός που αντιτίθεται σε έναν από τους κύριους πυλώνες για τη χρήση περιεκτών, που είναι η βέλτιστη αξιοποίηση των διαθέσιμων υπολογιστικών πόρων.

4.3.2 Βιβλιογραφική Ανασκόπηση

Η μελέτη της βιβλιογραφίας αποτελεί μια κρίσιμη και πολύπλοκη διαδικασία, αλλά παράλληλα απαραίτητη όταν αναζητείται η καταλληλότερη επιστημονικά και επιχειρησιακά προσέγγιση στην διαχείριση ενός προβλήματος. Η ομάδα διαχείρισης συστημάτων και δικτύων του ΕΛΚΕΘΕ έχοντας ως στόχο την ορθότερη και βέλτιστη αντιμετώπιση του προβλήματος που καλούνταν να επιλύσει αφιέρωσε τον απαιτούμενο χρόνο σε μία διεξοδική βιβλιογραφική ανασκόπηση.

Αρχικά η ομάδα διεξήγαγε έρευνα σχετικά με τις τάσεις της τεχνολογίας, ώστε να εντοπίσει τις πλέον επικαιροποιημένες και «ενεργές» λύσεις που είναι συμβατές με τις τρέχουσες τάσεις και τις μελλοντικές εξελίξεις. Παράλληλα, η ομάδα αναζήτησε μελέτες περίπτωσης, άρθρα και βιβλία που να περιγράφουν την διαχείριση παρόμοιων μετασχηματισμών υποδομών, χωρίς όμως να υπάρξει το αναμενόμενο πλήθος πληροφορίας συμβατής με το μέγεθος της υποδομής του φορέα. Τα δεδομένα αυτά θα μπορούσαν να παρέχουν πληροφορίες για τα πλεονεκτήματα, τα αδύναμα σημεία, τις δυνατότητες και τις προκλήσεις των διάφορων επιλογών (SWOT Analysis).

Για την υπερκέραση του παραπάνω προβλήματος η ομάδα διερεύνησε και αναζήτησε έρευνες και δημοσιεύσεις που αφορούν τις προδιαγραφές, την αποδοτικότητα, την ασφάλεια, την αξιοπιστία και την ευελιξία των πιθανών διαθέσιμων λύσεων. Αυτό προσέφερε μια ευρεία εικόνα των χαρακτηριστικών και των δυνατοτήτων των πιθανών λύσεων χωρίς όμως να μπορεί να εκτιμηθεί η συμβατότητα με τις υποδομές του φορέα.

Τέλος, η ομάδα εξέτασε την τεκμηρίωση και την υποστήριξη που παρέχεται για τις διάφορες λύσεις. Αυτό περιλάμβανε την εξέταση των επίσημων οδηγιών (documentation), των βοηθητικών οδηγιών (tutorials), των κοινοτήτων υποστήριξης (forums), των γνωσιακών βάσεων (knowledge base), καθώς και των πηγών εκπαίδευσης και κατάρτισης (trainings and seminars).

Μέσα από την διαδικασία της βιβλιογραφικής ανασκόπησης, η ομάδα συνέλεξε πληθώρα πληροφοριών, κατανόησε και αναγνώρισε τεχνολογίες που συμβαδίζουν

με τις ανάγκες των υποδομών, αλλά ταυτόχρονα διέκρινε έλλειψη σχετικά με την ύπαρξη αντίστοιχων μελετών περίπτωσης για ιδιωτικές υποδομές μικρού μεγέθους.

4.3.3 Τεχνική Αξιολόγηση

Η τεχνική ανάλυση και αξιολόγηση λύσεων είναι ένα κρίσιμο στάδιο στην εκτίμηση της υπάρχουσας κατάστασης. Αυτό περιλαμβάνει την εξέταση διάφορων διαθέσιμων λύσεων εγκατάστασης και διαχείρισης περιεκτών, όπως Docker, Kubernetes, OpenShift κ.λπ., και την εκτίμηση των επιπτώσεων, των απαιτήσεων, των πλεονεκτημάτων και των μειονεκτημάτων τους.

Επιπλέον, πρέπει να ληφθούν υπόψη και οι εξελίξεις στην τεχνολογία, όπως οι νεότερες πλατφόρμες και εργαλεία που ενδέχεται να παρέχουν αυξημένη απόδοση, καλύτερη ή ευκολότερη διαχείριση. Η εξέταση των λύσεων πρέπει να περιλαμβάνει και την εκτίμηση της δυνατότητας ενσωμάτωσης των λύσεων στην υφιστάμενη υποδομή, καθώς και των επιπτώσεων αυτής της ενσωμάτωσης στην λειτουργία και την απόδοση του συστήματος.

Τα βήματα που ακολουθήθηκαν από την ομάδα διαχείρισης συστημάτων και δικτύων περιλαμβάνουν την **κατανόηση της υφιστάμενης τεχνολογίας**, την **εξέταση των διαθέσιμων λύσεων**, την **εκτίμηση απαιτήσεων της κάθε λύσης** και την **αξιολόγηση της συμβατότητας τους με την υπάρχουσα υποδομή**.

4.3.3.1 Κατανόηση της υφιστάμενης τεχνολογίας

Όπως εκτενώς αναφέρθηκε και στην ανάλυση τεχνολογικών περιορισμών (βλ. 4.3.1.2), το ΕΛΚΕΘΕ στηρίζει κυρίως τις υποδομές του σε λογισμικά ανοιχτού κώδικα. Αυτά περιλαμβάνουν ανοιχτού κώδικα λειτουργικά συστήματα Linux, ανοιχτού κώδικα υπερεπώπτες εικονικών μηχανών (VMM), ανοιχτού κώδικα τείχη προστασίας και συσκευές δρομολόγησης, όπως και πληθώρα εφαρμογών ανοιχτού κώδικα για τις προς διάθεση υπηρεσίες (π.χ. web servers, βάσεις δεδομένων κ.λπ.). Η χρήση τεχνολογιών ανοιχτού κώδικα στο μέγιστο δυνατό βαθμό αποτελεί στρατηγική επιλογή του Υπολογιστικού Κέντρου (NOC) και του φορέα γενικότερα, δεδομένου και της δραστηριότητας του (ερευνητικό/ακαδημαϊκό ίδρυμα).

Πιο αναλυτικά, οι υποδομές διαχείρισης εικονικών μηχανών έχουν υλοποιηθεί αξιοποιώντας τις ανοιχτού κώδικα λύσεις oVirt της RedHat και Proxmox. Επίσης γίνεται απόλυτα περιορισμένη χρήση της ελεύθερης έκδοσης κλειστού κώδικα

λύσης VMware ESXi, καθώς δεν υπήρχε συμβατή εναλλακτική για την φιλοξενία προγενέστερων λειτουργικών συστημάτων της Microsoft (Windows 2000NT). Τα λειτουργικά συστήματα που χρησιμοποιούνται στην πληθώρα των εικονικών μηχανών είναι CentOS, Oracle Linux, Ubuntu και Debian.

Οι εικονικές μηχανές αυτές φιλοξενούν τις υπηρεσίες πάνω στις οποίες στηρίζονται τα ερευνητικά έργα και οι διοικητικές υπηρεσίες και περιλαμβάνουν mail services, file sharing services, web services, database services, monitoring services, ticketing services (helpdesk, booking κ.λπ.). Επιπλέον φιλοξενούνται και εικονικές μηχανές με λειτουργικό σύστημα FreeBSD που αξιοποιούνται ως routers, firewalls, VPN servers, files servers (TrueNAS, pfSense).

Οι υπάρχουσες δικτυακές υποδομές εντός των Datacenters κρίνονται επαρκείς ως προς την ταχύτητα, αξιοπιστία και διαθεσιμότητα τους. Βασίζονται σε πολλαπλές συνδέσεις των 40 και 10 Gbps που καλύπτουν τις απαιτήσεις των υποψήφιων προς υλοποίηση υποδομών, αφού προσφέρουν ικανοποιητικά υψηλό ρυθμό μετάδοσης, μικρό latency και υψηλή διαθεσιμότητα.

Οι διαθέσιμοι πόροι σε υλισμικό (εξυπηρετητές και συστήματα αποθήκευσης), αν και αναβαθμίζονται διαρκώς τα τελευταία έτη, λόγω της συνεχούς αυξανόμενης ζήτησης για νέες υπηρεσίες αλλά και για αναβάθμιση των υπαρχουσών υπηρεσιών αξιοποιούνται σε μεγάλο ποσοστό. Πρακτικά αυτό υποδεικνύει ότι οι διαθέσιμοι πόροι υλισμικού για τον μετασχηματισμό ή την προσθήκη νέων υποδομών που θα φιλοξενούν περιέκτες είναι πεπερασμένοι και δεικτικοί ως προς τις λύσεις που μπορούν να υλοποιηθούν.

4.3.3.2 Εξέταση διαθέσιμων λύσεων

Η ομάδα διαχείρισης συστημάτων και δικτύων του ΕΛΚΕΘΕ μέσα από μια διαρκή διαδικασία αναζήτησης και επιμόρφωσης κατέγραψε κάποιες πιθανές λύσεις, ανεξάρτητα πολυπλοκότητας, ανάγκης για τεχνογνωσία ή απαιτήσεων σε πόρους των λύσεων αυτών. Η συλλογή των πιθανών λύσεων επετεύχθη μέσα από αναζητήσεις στο διαδίκτυο, επικοινωνία με εξειδικευμένες στο αντικείμενο εταιρίες, μελέτη της σχετικής βιβλιογραφίας και παρακολούθηση συνεδρίων-σεμιναρίων.

Από το σύνολο των λύσεων που κατεγράφησαν αποκλείστηκαν, εξ αρχής, λύσεις που αφορούσαν την φιλοξενία περιεκτών σε δημόσια νεφοϋπολογιστικά

συστήματα, όπως Amazon Elastic Container Service (ECS), Amazon Elastic Kubernetes Service (EKS), Google Kubernetes Engine (GKE), Azure Kubernetes Service (AKS) καθώς δεν ικανοποιούσαν την βασική απαίτηση για φιλοξενία σε on premises υποδομές. Επιπλέον, λύσεις που περιείχαν τη χρήση LXC ή τη χρήση Kata Containers ή γενικότερα δεν υποστήριζαν εγγενώς Docker περιέκτες δεν εξετάστηκαν περαιτέρω καθώς δεν ικανοποιούσαν τις προτιμήσεις των χρηστών για υποστήριξη Docker περιεκτών. Παρομοίως αποκλείστηκε και η πλατφόρμα διαχείρισης περιεκτών Singularity, δεδομένου ότι έχει σχεδιαστεί ειδικά για περιέκτες σε περιβάλλοντα όπως High Performance Computing (HPC) και Machine Learning (ML). Παρόλα αυτά το Singularity θα μπορούσε να εξεταστεί στο μέλλον ως μια λύση για την διαχείριση και την εκτέλεση επιστημονικών μοντέλων από τις επιστημονικές ομάδες του ΕΛΚΕΘΕ που ασχολούνται με τα HPCs.

Οι λύσεις που κατεγράφησαν προς μελέτη παρατίθενται ακολούθως και ομαδοποιούνται σε τρεις κατηγορίες. Στις **λύσεις που οι περιέκτες εκτελούνται σε ένα μηχάνημα** – φυσικό ή εικονικό, στις **λύσεις που οι περιέκτες εκτελούνται σε συστοιχίες μηχανών** και στις **λύσεις που οι περιέκτες εκτελούνται είτε σε μία μηχανή είτε σε συστοιχίες μηχανών αλλά η διαχείριση τους επιτυγχάνεται μέσω κάποιας πλατφόρμας ενορχήστρωσης**. Η τρίτη λύση δύναται να περιέχει τον συνδυασμό της δεύτερης λύσης μαζί με κάποιο γραφικό περιβάλλον διαχείρισης.

Περιέκτες σε ένα μηχάνημα

Docker Engine/Docker Compose: Η λύση αυτή αφορά την δυνατότητα εκτέλεσης ενός ή περισσότερων περιεκτών σε ένα μεμονωμένο υπολογιστικό μηχάνημα και μπορεί εύκολα να αξιοποιήσει την διαθέσιμη υποδομή εικονικών μηχανών (IaaS) του ΕΛΚΕΘΕ. Πιο συγκεκριμένα το Docker Compose μπορεί να καλύψει την ανάγκη εκτέλεσης πολλών περιεκτών ως μια ενιαία εφαρμογή ενώ παράλληλα καθιστά εύκολη την διαχείριση των παρεχόμενων πόρων και διασφαλίζει την απομόνωση τους αφού κάθε περιέκτης ή ομάδα αυτών εκτελούνται σε διαφορετικό μηχάνημα. Το Docker αξιοποιήθηκε από την ομάδα διαχείρισης συστημάτων και δικτύων του ΕΛΚΕΘΕ ως αρχική πλατφόρμα εκτέλεσης καθώς και για περαιτέρω εξοικείωση με την τεχνολογία των περιεκτών. Η αδυναμία όμως του Docker να εκτείνεται σε συστοιχίες υπολογιστικών μηχανών στάθηκε καταλυτικός παράγοντας ώστε να μην επιλεγεί ως μια λύση για τη διαχείριση περιεκτών στον

Φορέα, αλλά επιβεβαιώθηκε ως μία επαρκής λύση για την γρήγορη και ασφαλή εκτέλεση μεμονωμένων εφαρμογών που δεν απαιτούν ελαστικότητα.

Περιέκτες που εκτελούνται σε συστοιχίες μηχανών

Docker Swarm: Το Docker Swarm δίνει τη δυνατότητα δημιουργίας ενός σμήνους (swarm), δηλαδή μιας συλλογής κόμβων (nodes – υπολογιστές που εκτελούν Docker) οι οποίοι συνεργάζονται μεταξύ τους μέσω δικτυακής επικοινωνίας. Αυτοί οι κόμβοι μπορεί να είναι manager κόμβοι οι οποίοι διαχειρίζονται την κατανομή των εργασιών στο σμήνος ή worker κόμβοι που εκτελούν τις ίδιες τις εργασίες. Το Docker Swarm αποτελεί την προεπιλεγμένη μηχανή ενορχήστρωσης του Docker και παρέχει δυνατότητες εύκολης κλιμάκωσης, αυτόματης διανομής φόρτων εργασίας και διαχείρισης υπηρεσιών σε ευρύτερη κλίμακα. Αν και είναι μια πλατφόρμα ενορχήστρωσης που προσφέρει εξαιρετική απλότητα και ευκολία χρήσης, εντούτοις έχει μερικά μειονεκτήματα που δεν το καθιστούν ως την βέλτιστη επιλογή. Τέτοια αποτελούν η περιορισμένη αυτοματοποίηση, καθώς δεν διαθέτει δυνατότητες αυτόματης ανακατανομής των περιεκτών και αυτόματης κλιμάκωσης (auto scaling), οι μειωμένες δυνατότητες διαχωρισμού και απομόνωσης των εργασιών ανά ομάδες χρηστών και η άρρηκτη εξάρτηση του από το Docker.

Kubernetes: Το Kubernetes, γνωστό και ως K8s, είναι ένα ανοιχτού κώδικα σύστημα που αυτοματοποιεί τη διαχείριση, την κλιμάκωση και την εναπόθεση των εφαρμογών σε περιέκτες. Αναπτύχθηκε αρχικά από την Google και σήμερα το διαχειρίζεται το Cloud Native Computing Foundation (CNCF). Αυτοματοποιεί τη διαχείριση περιεκτών, περιλαμβανομένης της διάθεσης, της δικτύωσης και της κλιμάκωσης των εφαρμογών. Μερικές από τις κυριότερες λειτουργίες του Kubernetes περιλαμβάνουν, την διαχείριση εφαρμογών (stateful, stateless και data-processing workloads), την αυτόματη κλιμάκωση, την ευελιξία δικτύωσης (Network Policies – επικοινωνία των pods) και το self-healing. Το Kubernetes είναι μια ισχυρή και περίπλοκη πλατφόρμα και η εκμάθησή του μπορεί να είναι απαιτητική. Λόγω αυτής της πολυπλοκότητας και της πληθώρας ρυθμίσεων που απαιτεί, προϋποθέτει εκπαίδευση και εμπειρία για την διαχείριση του και χωρίς κάποιο επιπρόσθετο περιβάλλον διαχείρισης καθίσταται δύσχρηστο και δυσνόητο για τους υποψήφιους χρήστες.

Ελαφριές εκδόσεις του Kubernetes (K3s, MicroK8s κ.λπ.) : Οι εκδόσεις αυτές, ως επί το πλείστον, διαθέτουν όλα τα βασικά χαρακτηριστικά του Kubernetes, αλλά είναι βελτιστοποιημένες είτε για edge και IoT, είτε για πιο ελαφριές εφαρμογές. Είναι ιδανικές για περιβάλλοντα όπου οι πόροι είναι περιορισμένοι ή προτιμάται η διαχείριση πολλαπλών Kubernetes clusters έναντι μιας μεγαλύτερης Kubernetes υποδομής. Όπως και το Kubernetes χρειάζεται κάποιο επιπρόσθετο περιβάλλον διαχείρισης ώστε να είναι πιο προσιτό στους υποψήφιους χρήστες.

Περιέκτες συνδυαστικά με πλατφόρμα διαχείρισης

Apache Mesos: Είναι μια ανοικτού κώδικα πλατφόρμα διαχείρισης πόρων και χρονοπρογραμματισμού εργασιών, που προσφέρει πολύ καλές δυνατότητες κλιμάκωσης. Συχνά χρησιμοποιείται μαζί με το Marathon, ένα framework που παρέχει δυνατότητες διαχείρισης περιεκτών. Παρόλα αυτά η βασική λειτουργία του Mesos αφορά την υποστήριξη μιας ποικιλίας εφαρμογών όπως των Hadoop, Spark και Kafka, κάτι που δεν το προκρίνει ως επιλογή δεδομένου ότι ανταγωνιστικές πλατφόρμες εστιάζουν αποκλειστικά και μόνο στην διαχείριση περιεκτών. Επιπρόσθετα η τεκμηρίωση και η υποστήριξη της κοινότητας σχετικά με την διαχείριση περιεκτών κρίθηκε περιορισμένη και παρόλο που το Apache Mesos παρέχει ένα ευέλικτο πλαίσιο για τη διαχείριση πόρων, τείνει να είναι πιο δύσκολο στη ρύθμιση και τη διαχείριση σε σχέση με άλλα συστήματα όπως το Kubernetes. Η αρχιτεκτονική του Mesos είναι αρκετά πολύπλοκη και μπορεί να απαιτήσει μεγαλύτερο βάθος γνώσης σε σχέση με άλλες λύσεις, ιδίως σε περιβάλλοντα παραγωγής.

Nomad HashiCorp: Το Nomad είναι ένα εργαλείο για τη διαχείριση του κύκλου ζωής των εφαρμογών, που παρέχεται από την HashiCorp, την ίδια εταιρεία που αναπτύσσει το Vagrant, το Packer, το Terraform και άλλα δημοφιλή εργαλεία DevOps. Έχει επίσης τη δυνατότητα να διαχειρίζεται διάφορους τύπους εργασιών, όπως εφαρμογές Docker, απομονωμένες εφαρμογές (chroot), και VM μέσω του QEMU. Ωστόσο, αν και το Nomad είναι εξαιρετικά ισχυρό και ευέλικτο, δεν παρέχει κάποιες από τις πιο προηγμένες λειτουργίες του Kubernetes, όπως service discovery, load balancing και διαχείριση secrets. Παρόλο που οι λειτουργίες αυτές μπορούν να παρέχονται μέσω άλλων εργαλείων της HashiCorp όπως το Consul και το Vault, απορρίπτεται σαν λύση γενικότερα αφού ανταγωνιστικές λύσεις προσφέρουν αυτές τα εργαλεία εγγενώς.

OpenStack: Το OpenStack είναι μια ανοιχτού κώδικα πλατφόρμα λογισμικού που παρέχει λύσεις για τη δημιουργία και διαχείριση ιδιωτικών και δημόσιων νεφών. Προσφέρει ένα ευέλικτο περιβάλλον για την εκτέλεση εικονικών μηχανών, τον χειρισμό αποθηκευτικού χώρου και τη διαχείριση δικτύων. Παρά τα πλεονεκτήματά του, η ομάδα διαχείρισης συστημάτων και δικτύων δεν επέλεξε το OpenStack ως λύση νεφούπολογιστικής υποδομής. Ένας από τους λόγους είναι η πολυπλοκότητα της εγκατάστασης και της διαχείρισης του. Η παραμετροποίηση και η συντήρηση του συστήματος απαιτούν τεχνικές γνώσεις και εμπειρία, καθώς οι πολλές συνιστώσες του OpenStack απαιτούν προσαρμογές και ρυθμίσεις. Ένας άλλος λόγος ήταν η απαιτούμενη υποδομή για την εκτέλεση του OpenStack. Η πλατφόρμα απαιτεί ένα σύνολο φυσικών υπολογιστικών πόρων, όπως εξυπηρετητές, αποθηκευτικούς χώρους και δίκτυα, προκειμένου να λειτουργήσει βέλτιστα. Η ανάπτυξη και η διατήρηση μιας τέτοιας υποδομής δεν ήταν εφικτή υπό τους όρους του κόστους και του χρόνου. Επιπλέον, η πολυπλοκότητα του OpenStack μπορεί να οδηγήσει σε δυσκολίες στην επέκταση και την ενσωμάτωση νέων τεχνολογιών ή υπηρεσιών. Οι αναβαθμίσεις και οι ενημερώσεις μπορεί να απαιτούν σημαντικό χρόνο και προσπάθεια, και η ανεπάρκεια πόρων ή η έλλειψη τεχνογνωσίας μπορεί να οδηγήσει σε προβλήματα και διακοπές των υπηρεσιών. Όλα τα παραπάνω αποτέλεσαν μεγάλη πρόκληση για το μέγεθος του ΕΛΚΕΘΕ και οδήγησαν στην απόρριψη της λύσης προσωρινά, αλλά πιθανότατα χρήζει επανεξέτασης αν διαμορφωθούν καταλληλότερες συνθήκες στο μέλλον.

Portainer: Το Portainer είναι μια ενοποιημένη διεπαφή για τη διαχείριση Docker περιεκτών και από την έκδοση 2.0 και μετά υποστηρίζει και τη διαχείριση των περισσότερων διανομών Kubernetes, είτε αυτές είναι on premises είτε αυτές προσφέρονται από παρόχους υπηρεσιών νέφους. Η γραφική του διεπαφή προσφέρει ένα ευέλικτο και εύχρηστο εργαλείο διαχείρισης και μπορεί να αξιοποιηθεί για την απλοποίηση εργασιών διαχείρισης και την παρακολούθηση (monitoring) των περιεκτών, των πόρων και των συστοιχιών. Παρότι το Portainer παρουσιάζεται ως μια ενδιαφέρουσα προς εξέταση λύση, η αδυναμία δημιουργίας και διαμόρφωσης συστοιχιών Kubernetes, CRDs και namespaces συνδυαστικά με την έλλειψη διαλειτουργικότητας με IaaS υποδομές, το απομακρύνουν από τις προκρινθείσες λύσεις.

Kubernetes: Το Kubernetes είναι μια ανοιχτού κώδικα πλατφόρμα που παρέχει πλήρη διαχείριση Kubernetes, υποστηρίζει διάφορες εφαρμογές και περιβάλλοντα και παρέχει ενσωματωμένες λειτουργίες DevOps που βοηθούν στην αυτοματοποίηση διαδικασιών ανάπτυξης εφαρμογών. Το Kubernetes διαθέτει μία πλούσια γραφική διεπαφή χρήστη που καθιστά ευκολότερη την διαχείριση των Kubernetes clusters, ενώ υποστηρίζει εγγενώς την πολυχρηστικότητα (multitenancy), κάτι ιδιαίτερα σημαντικό σε περιβάλλοντα που απαιτούν διαχωρισμό πόρων και χρηστών. Η αδυναμία του Kubernetes να διαλειτουργήσει με IaaS υποδομές εγγενώς, είτε με την χρήση κάποιου πρόσθετου εργαλείου (plugin / module / addon), το καθιστούν μη επιλέξιμη λύση.

Rancher: Το Rancher είναι μια ανοιχτού κώδικα πλατφόρμα διαχείρισης περιεκτών που σχεδιάστηκε για να επιτρέψει σε οργανισμούς να εκτελούν και να διαχειρίζονται συστοιχίες Kubernetes σε οποιοδήποτε περιβάλλον. Επιπλέον, υποστηρίζει όλες τις κύριες διανομές Kubernetes και επιτρέπει στους χρήστες να διαχειρίζονται όλες τις λειτουργίες τους Kubernetes από ένα κεντρικό σημείο ελέγχου. Το Rancher παρέχει μια κεντρική, γραφική διεπαφή χρήστη για τη διαχείριση όλων των συστοιχιών Kubernetes και μπορεί να διαχειριστεί πολλαπλές συστοιχίες Kubernetes, είτε είναι on-premises, είτε στο νέφος, είτε σε μια υβριδική διάταξη. Επιπροσθέτως, υποστηρίζει μία ποικιλία IaaS πλατφορμών, συμπεριλαμβανομένων AWS, Azure, Google Cloud, VMware και Bare Metal. Άξια αναφοράς και μελέτης είναι η ανοιχτού κώδικα HCI πλατφόρμα Harvester που αξιοποιεί Bare Metal ή άλλες IaaS υποδομές και ενσωματώνεται πλήρως στη διαχειριστική πλατφόρμα του Rancher. Το Harvester συνδυάζει τις λειτουργίες του Kubernetes και του KubeVirt για να παρέχει μια πλατφόρμα που μπορεί να διαχειριστεί τόσο περιέκτες όσο και εικονικές μηχανές. Αυτό σημαίνει ότι μπορεί να εκτελεί παράλληλα περιέκτες και εικονικές μηχανές στην ίδια συστοιχία, δίνοντάς περισσότερη ευελιξία στην διαχείριση των υποδομών. Το πλήθος των δυνατοτήτων του Rancher το καθιστούν μια πολλά υποσχόμενη λύση, η οποία προκρίνεται προς εξέταση. Ο δε συνδυασμός του με την πλατφόρμα διαχείρισης πόρων Harvester, το ξεχωρίζουν ως μία ακόμα πιο ελκυστική και ολοκληρωμένη διαμόρφωση υποδομών για μικρούς οργανισμούς

OKD: Το OKD, που αρχικά αναφερόταν στο "Origin Community Distribution of Kubernetes", είναι η έκδοση κοινότητας του Red Hat OpenShift, της πλατφόρμας

διαχείρισης Kubernetes που παρέχει η Red Hat. Το OKD είναι ανοικτού κώδικα και παρέχεται δωρεάν, επιτρέποντας στους χρήστες να δημιουργούν και να διαχειρίζονται περιβάλλοντα Kubernetes χωρίς να απαιτείται η αγορά συνδρομής Red Hat OpenShift. Η έκδοση ανοικτού κώδικα της OpenShift, προσφέρει ευελιξία, αφού μπορεί να εγκατασταθεί σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των εικονικών μηχανών, των bare metal εξυπηρετητών και των δημόσιων νεφών, υποστηρίζει την αυτόματη κλιμάκωση, την ευκολία διαχείρισης και την υποστήριξη της συνεχούς παράδοσης και συνεχούς ενσωμάτωσης (CI/CD). Το ιδιαίτερα ελκυστικό χαρακτηριστικό του OKD είναι η δυνατότητα ενσωμάτωσης τους σε άλλες πλατφόρμες, όπως το oVirt που παρέχει μια αυτοματοποιημένη διαχείριση και παραμετροποίηση των IaaS υποδομών στις οποίες φιλοξενείται. Οι ήδη εγκατεστημένες υποδομές στον φορέα, επιτάσσουν την εξέταση της λύσης αυτής.

Σύμφωνα με τα παραπάνω η ομάδα διαχείρισης συστημάτων και δικτύων κατέληξε να εξετάσει, εγκαθιστώντας στις υπάρχουσες υποδομές του Φορέα και δοκιμάζοντας στην πράξη, τις εξής προκρινθείσες λύσεις:

- Rancher, με ή χωρίς Harvester, με διάφορες διανομές Kubernetes
- OKD, αξιοποιώντας την δυνατότητα του για ενσωμάτωση στο oVirt

Παράλληλα και λόγω του ότι οι δύο προκρινθείσες λύσεις στηρίζονται στην διαλειτουργικότητα με συγκεκριμένες IaaS υποδομές, διερευνήθηκαν και οι εναλλακτικές λύσεις που θα μπορούσαν να εξεταστούν ως πλατφόρμες διαχείρισης ανεξάρτητων Kubernetes συστοιχιών, βασιζόμενες στις υπάρχουσες IaaS υποδομές. Οι εναλλακτικές λύσεις που προκρίθηκαν, σύμφωνα με τις δυνατότητες που παρέχουν και την ευκολία εγκατάστασης και διαχείρισης είναι οι εξής:

- Portainer, συμπληρωματικά με διάφορες διανομές Kubernetes
- Kubesphere, με ή χωρίς διάφορες διανομές Kubernetes

Οι λύσεις, συνολικά, επιλέχθηκαν και με γνώμονα τις διεθνείς τάσεις στο χώρο των νεφούπολογιστικών τεχνολογιών που ακολουθούν υλοποιήσεις που βασίζονται κατά κύριο λόγο σε συστοιχίες Kubernetes.

4.3.3.3 Εκτίμηση απαιτήσεων της κάθε λύσης

Η ομάδα διαχείρισης συστημάτων και δικτύων μελέτησε πιο διεξοδικά τις ανάγκες για υλικοτεχνική υποστήριξη, τις ανάγκες για εκπαίδευση και τις δυνατότητες διαχείρισης και αποδοτικότητας των λύσεων, πριν προχωρήσει στην δοκιμαστική υλοποίηση τους. Αρχικά εξέτασε τις ανάγκες για διανομές Kubernetes, δεδομένου ότι όλες οι λύσεις περιλαμβάνουν, άμεσα ή έμμεσα, μία ή περισσότερες διανομές Kubernetes.

Διανομές Kubernetes (Kubernetes, K3s, RKE, MicroK8s)

Οι ανάγκες και οι δυνατότητες για όλες τις διανομές Kubernetes που θα εξεταστούν από την ομάδα διαχείρισης συστημάτων και δικτύων μπορούν να ομαδοποιηθούν ως εξής:

Υλικοτεχνική Υποστήριξη: Το κλασικό Kubernetes, το K3s και το RKE υποστηρίζουν ευρεία γκάμα υλικού και μπορούν να εκτελεστούν σε on-premises εξυπηρετητές ή σε υποδομές νέφους. Το MicroK8s είναι πιο ευέλικτο, καθώς μπορεί να εκτελεστεί σε εξυπηρετητές, προσωπικούς υπολογιστές ακόμη και σε IoT συσκευές.

Ανάγκες για Εκπαίδευση: Το κλασικό Kubernetes και το RKE απαιτούν σημαντική κατανόηση των βασικών εννοιών και τεχνολογιών του Kubernetes. Από την άλλη πλευρά, το K3s και το MicroK8s είναι πιο απλοποιημένα, καθιστώντας τα πιο προσβάσιμα για νέους χρήστες ή για χρήστες με περιορισμένη εμπειρία στο Kubernetes.

Δυνατότητες Διαχείρισης και Αποδοτικότητας: Το κλασικό Kubernetes και το RKE προσφέρουν ισχυρές δυνατότητες διαχείρισης, συμπεριλαμβανομένης της αυτόματης κλιμάκωσης, της ισορροπίας φόρτου και της διαχείρισης υπηρεσιών. Το K3s και το MicroK8s, ενώ παρέχουν βασικές δυνατότητες διαχείρισης, είναι πιο περιορισμένα σε σχέση με την αποδοτικότητα και την κλιμάκωση σε μεγάλα περιβάλλοντα.

Rancher, με ή χωρίς Harvester, με διάφορες διανομές Kubernetes

Το Rancher είναι μια πλατφόρμα διαχείρισης και αυτοματοποίησης για την ανάπτυξη και τη λειτουργία περιβαλλόντων περιεκτών. Όπως κάθε λύση

λογισμικού, έχει ορισμένες απαιτήσεις και παράγοντες που πρέπει να ληφθούν υπόψη.

Υλικοτεχνική υποστήριξη: Το Rancher μπορεί να εκτελεστεί σε φυσικούς ή εικονικούς διακομιστές και απαιτεί μια ομάδα υπολογιστικών πόρων για τη λειτουργία του. Οι ακριβείς απαιτήσεις εξαρτώνται από τον αριθμό των κόμβων και την κλίμακα του περιβάλλοντος που θα διαχειριστεί το Rancher. Είναι σημαντικό να διαθέτει επαρκείς υπολογιστικούς πόρους (επεξεργαστές, μνήμη, αποθηκευτικό χώρο) για να διασφαλιστεί η ομαλή λειτουργία του.

Ανάγκες για εκπαίδευση: Η χρήση του Rancher απαιτεί μια κατανόηση των βασικών αρχών των περιεκτών και των εργαλείων που χρησιμοποιούνται στο συγκεκριμένο περιβάλλον. Εκπαιδευτικοί πόροι, όπως τεκμηρίωση, οδηγοί χρήσης και διαδικτυακά μαθήματα, που μπορούν να υποστηρίξουν την εκμάθηση, την υλοποίηση και την χρήση του, διατίθενται ελεύθερα τόσο στον επίσημο ιστότοπο του Rancher, όσο και ευρύτερα στο διαδίκτυο.

Δυνατότητες διαχείρισης και αποδοτικότητας: Το Rancher παρέχει πλούσιες δυνατότητες διαχείρισης περιβαλλόντων περιεκτών, συμπεριλαμβανομένης της διαμόρφωσης, της κλιμάκωσης, της παρακολούθησης και της αυτοματοποίησης και φαίνεται να αποτελεί μια ισχυρή λύση για τη διαχείριση περιβαλλόντων περιεκτών.

Το Harvester είναι ένα ενοποιημένο σύστημα υπολογιστικής υποδομής ανοιχτού κώδικα (HCI) που φέρνει την ευκολία της διαχείρισης Kubernetes σε επίπεδο υποδομής. Συνδυάζει εργαλεία ανοιχτού κώδικα για αποθήκευση, δίκτυο και εικονικοποίηση και τα πακετάρει σε ένα διαχειρίσιμο και εύχρηστο περιβάλλον.

Υλικοτεχνική υποστήριξη: Το Harvester απαιτεί σύγχρονο υλικό υπολογιστή που υποστηρίζει την εικονικοποίηση KVM (Kernel-based Virtual Machine). Ωστόσο, λόγω της αρχιτεκτονικής του Kubernetes, μπορεί να εγκατασταθεί και να λειτουργήσει σε ένα σύνολο από διακομιστές, δημιουργώντας ένα συμπαγές και ευρύ δίκτυο.

Ανάγκες για εκπαίδευση: Όσοι έχουν ήδη εμπειρία με το Kubernetes θα βρουν το Harvester ευθυγραμμισμένο με τη γνώση τους. Σε αντίθετη περίπτωση, η εκμάθηση του Kubernetes και της διαχείρισης της υποδομής μέσω του Harvester απαιτεί εκπαίδευση και χρόνο.

Δυνατότητες διαχείρισης: Το Harvester προσφέρει μία κεντρική διαδικτυακή εφαρμογή για διαχείριση της υποδομής HCI. Το σύνολο των λειτουργιών, από τη δημιουργία και τη διαχείριση εικονικών μηχανών μέχρι τη διαμόρφωση του δικτύου και την αποθήκευση, μπορούν να εκτελεστούν μέσα από αυτό το περιβάλλον.

Αποδοτικότητα: Όπως και οι περισσότερες λύσεις Kubernetes, το Harvester αυξάνει την αποδοτικότητα μέσω της αυτοματοποίησης και της ευελιξίας. Η υποστήριξη για απομόνωση πόρων, αυτόματη εξισορρόπηση φόρτου και η δυνατότητα μετατροπής εικονικών μηχανών σε περιέκτες παρέχουν ισχυρές ευκαιρίες βελτιστοποίησης.

Portainer, συμπληρωματικά με διάφορες διανομές Kubernetes

Το Portainer είναι μια ελαφριά πλατφόρμα διαχείρισης για περιβάλλοντα περιεκτών, που στοχεύει στην απλότητα και την ευκολία χρήσης.

Υλικοτεχνική Υποστήριξη: Το Portainer μπορεί να εκτελεστεί σε διάφορα περιβάλλοντα, καθώς είναι μια πλατφόρμα ανοιχτού κώδικα που υποστηρίζει τόσο Docker όσο και συστοιχίες Kubernetes. Μπορεί να εκτελεστεί σε on-premises εξυπηρετητές, σε νεφούπολογιστικά περιβάλλοντα καθώς και σε edge περιβάλλοντα.

Ανάγκες για Εκπαίδευση: Το Portainer είθισται να επιλέγεται για την ευκολία χρήσης του, καθώς προσφέρεται με μία φιλική, προς τον χρήστη, γραφική διεπαφή που καθιστά πιο προσβάσιμες τις πολύπλοκες λειτουργίες της διαχείρισης περιεκτών. Επιπλέον, η κοινότητα του Portainer παρέχει εκτενείς πηγές εκμάθησης και υποστήριξης.

Δυνατότητες Διαχείρισης και Αποδοτικότητας: Το Portainer προσφέρει ευρεία γκάμα δυνατοτήτων διαχείρισης, όπως τη δυνατότητα διαχείρισης περιεκτών, εικόνων περιεκτών, δικτύων και αποθηκευτικών μονάδων (VC), καθώς και τη δυνατότητα να χειρίζεται υπηρεσίες Kubernetes όπως pods, services και deployments. Αν και το Portainer είναι επαρκώς αποδοτικό για την καθημερινή διαχείριση περιεκτών, η απόδοση της εφαρμογής εξαρτάται από τις υποδομές στις οποίες εκτελείται.

Kubernetes, με ή χωρίς διάφορες διανομές Kubernetes

Το Kubesphere είναι μια πλατφόρμα διαχείρισης Kubernetes που προσφέρει εύκολη διαχείριση και ολοκληρωμένη λειτουργικότητα, όπως η ενορχήστρωση πολλαπλών συστοιχιών, η πολυχρηστικότητα (multitenancy), η συνεχής ενσωμάτωση και διανομή (CI/CD), η παρακολούθηση (monitoring) και η καταγραφή (logging), σε μια ενιαία πλατφόρμα.

Υλικοτεχνική Υποστήριξη: Το Kubesphere είναι μια ανοιχτού κώδικα πλατφόρμα, που είναι σχεδιασμένη να εκτελείται, είτε απευθείας σε οποιαδήποτε διανομή Kubernetes, είτε σε διανομές Linux. Αυτό σημαίνει ότι μπορεί να εκτελείται σε μια ευρεία γκάμα υλικοτεχνικών περιβαλλόντων, είτε αυτά είναι on-premises, είτε νεφούπολογιστικά περιβάλλοντα.

Ανάγκες για Εκπαίδευση: Ως μια από τις πλέον διαδεδομένες πλατφόρμες διαχείρισης Kubernetes, το Kubesphere διαθέτει από πιο απλές έως και αρκετά προηγμένες λειτουργίες, οι οποίες απαιτούν κάποια εξοικείωση και εκπαίδευση για την πλήρη αξιοποίησή τους. Ωστόσο, διαθέτει εκτενή τεκμηρίωση και μια ενεργή κοινότητα για την υποστήριξη των νέων χρηστών.

Δυνατότητες Διαχείρισης και Αποδοτικότητας: Το Kubesphere προσφέρει μια πλήρη, ενοποιημένη πλατφόρμα για τη διαχείριση των συστοιχιών Kubernetes, καθώς και πολλές ενσωματωμένες λειτουργίες, όπως πολυχρηστικότητα, πολιτικές δικτύου, διαχείριση αποθηκευτικών πόρων, παρακολούθηση (monitoring) και σύστημα ειδοποιήσεων (alerting). Η αποδοτικότητα της πλατφόρμας εξαρτάται από τις υποδομές στις οποίες εκτελείται, αν και η αρχιτεκτονική του Kubesphere, σύμφωνα με την τεκμηρίωση του [68], σχεδιάστηκε για να μεγιστοποιεί την αποδοτικότητα και την απόδοση των εφαρμογών που φιλοξενεί.

OKD, αξιοποιώντας την δυνατότητα του για ενσωμάτωση στο oVirt

Το OKD είναι η έκδοση κοινότητας του OpenShift, της πλατφόρμας διαχείρισης περιεκτών της Red Hat. Αυτή η ανοιχτού κώδικα πλατφόρμα έχει υλοποιηθεί για να απλοποιήσει την διανομή, την κλιμάκωση και την ασφάλεια των εφαρμογών στο νέφος, με τη δυνατότητα εγγενούς ενσωμάτωσης με άλλα εργαλεία όπως το oVirt για την εικονική διαχείριση υπολογιστικών πόρων.

Υλικοτεχνική Υποστήριξη: Το OKD είναι μια διανομή του Kubernetes και, ως εκ τούτου, διαθέτει ευρεία υποστήριξη για διάφορα συστήματα υλικού. Επίσης, επειδή είναι η κοινοτική έκδοση του OpenShift, παρέχεται υποστήριξη και τεκμηρίωση από

την κοινότητα, αλλά στην περίπτωση που απαιτηθεί υπάρχει η δυνατότητα επίσημης υποστήριξης από την Red Hat .

Ανάγκες για Εκπαίδευση: Επειδή το OKD είναι βασισμένο στο Kubernetes, η τεχνογνωσία και η εκπαίδευση σχετικά με το Kubernetes είναι πολύτιμες. Ωστόσο, απαιτείται επιπλέον εκμάθηση για την κατανόηση των ειδικών χαρακτηριστικών και δυνατοτήτων του OKD, όπως η ολοκληρωμένη διαχείριση υπηρεσιών.

Δυνατότητες Διαχείρισης: Το OKD προσφέρει πολλές εξελιγμένες λειτουργίες διαχείρισης που πηγάζουν από το OpenShift, όπως ενσωματωμένη διαχείριση υπηρεσιών, εξελιγμένη διαχείριση υποδομών και εργαλεία για CI/CD.

Αποδοτικότητα: Η αποδοτικότητα του OKD εξαρτάται από τον τρόπο εφαρμογής και διαμόρφωσης του, καθώς και από το υποκείμενο υλισμικό και την συνολικότερη υποδομή. Σε γενικές γραμμές, το OKD προσπαθεί να μεγιστοποιήσει την αποδοτικότητα και την ευελιξία του Kubernetes, παρέχοντας πρόσθετες λειτουργίες και εργαλεία.

Ενσωμάτωση με oVirt: Το OKD μπορεί να ενσωματωθεί εγγενώς με το oVirt, μια ισχυρή πλατφόρμα διαχείρισης εικονικών μηχανών ανοιχτού κώδικα. Αυτό σημαίνει ότι μπορεί να χρησιμοποιεί το oVirt για να διαχειριστεί την υποδομή στην οποία φιλοξενείται, ενώ παράλληλα το OKD χρησιμοποιείται για τη διαχείριση των περιεκτών που φιλοξενούν τις εφαρμογές. Η ενσωμάτωση του OKD με το oVirt μπορεί να προσφέρει ευελιξία, αυτοματοποίηση και κλιμάκωση, ενώ εκμεταλλεύεται τις ανοιχτές προδιαγραφές και την κοινότητα του ανοιχτού κώδικα.

4.3.3.4 Αξιολόγηση της Συμβατότητας

Η επιλογή της κατάλληλης τεχνολογικής λύσης που θα συμβαδίζει αρμονικά με την υπάρχουσα υποδομή, είναι μία από τις σημαντικότερες προκλήσεις που αντιμετωπίζουν οι οργανισμοί. Η αξιολόγηση της συμβατότητας των διαφόρων λύσεων απαιτεί μια σαφή κατανόηση τόσο των τεχνικών απαιτήσεων, των πόρων και των δυνατοτήτων που προσφέρει κάθε λύση, όσο και άλλων κριτηρίων όπως η ευκολία ενσωμάτωσης, οι δυνατότητες αναβάθμισης, η διαθεσιμότητα υποστήριξης και η σχέση κόστους-ωφέλειας. Στα πλαίσια αυτά, ο καθορισμός συγκεκριμένων κριτηρίων, αποτελεί το πρώτο απαραίτητο βήμα για την επιλογή της πλέον κατάλληλης λύσης.

Η ομάδα διαχείρισης συστημάτων και δικτύων του ΕΛΚΕΘΕ, με γνώμονα την παραπάνω συλλογιστική, συνέταξε μία λίστα από κριτήρια για την αξιολόγηση της συμβατότητας των λύσεων που προκρίθηκαν προς εξέταση. Τα κριτήρια αυτά, περιλαμβάνουν την αξιολόγηση της συμβατότητας ως προς τις τεχνικές απαιτήσεις, την ευκολία ενσωμάτωσης, τις δυνατότητες αναβάθμισης, τη διαθέσιμη τεκμηρίωση, την δυνατότητα επίσημης υποστήριξης, την ωριμότητα της τεχνολογίας, καθώς και την προαπαιτούμενη τεχνογνωσία. Η αξιολόγηση του κόστους, που τυχόν απαιτείται για την υλοποίηση, αναλύεται ξεχωριστά, δεδομένου ότι ο οικονομικός παράγοντας αποτελεί δεσμευτικό κριτήριο αναφορικά με την επιλογή της όποιας λύσης.

Οι λύσεις που προάχθεισαν προς εξέταση μετά την καταγραφή και εκτίμηση του συνόλου των διαθέσιμων λύσεων είναι οι πλατφόρμες Rancher συνδυαστικά με Harvester και OKD συνδυαστικά με oVirt. Επιπρόσθετα στην εξέταση συμπεριλαμβάνονται οι πλατφόρμες Portainer και Kubesphere, ως εναλλακτικές λύσεις. Οι τέσσερις αυτές λύσεις θα αξιολογηθούν αναλυτικότερα σύμφωνα με τα κριτήρια που όρισε η ομάδα διαχείρισης συστημάτων και δικτύων.

4.3.3.4.1 Τεχνικές απαιτήσεις

Η συμβατότητα ως προς τις τεχνικές απαιτήσεις περιλαμβάνει την εξέταση παραμέτρων που αφορούν το υλισμικό, τη δικτύωση, τις απαιτήσεις ασφαλείας και τη συμβατότητα με υπάρχουσες εφαρμογές και υπηρεσίες. Όλες οι λύσεις μπορούν να θεωρηθούν συμβατές με την υπάρχουσα υποδομή και τις φιλοξενούμενες υπηρεσίες, αφού ικανοποιούν τους τεχνολογικούς περιορισμούς που έχουν τεθεί (βλ. 4.3.1.2) όπως η συμβατότητα με συστήματα Linux και με τις ήδη εγκατεστημένες υπηρεσίες, η ενσωμάτωση τους σε υποδομές IaaS που διαθέτει ο Φορέας και η εφαρμογή πρακτικών ασφαλείας όπως η χρήση επίσημων αποθετηρίων εικόνων περιεκτών και η απόδοση ελάχιστων δικαιωμάτων.

4.3.3.4.2 Ευκολία ενσωμάτωσης

Σύμφωνα με την υπάρχουσα υποδομή η οποία περιλαμβάνει φυσικούς εξυπηρετητές και νεφούπολογιστικές υποδομές εικονικών μηχανών (IaaS), οι μέθοδοι για την ενσωμάτωση των λύσεων αυτών βασίζονται είτε στην εγκατάσταση τους ως υπηρεσίες στην IaaS υποδομή, είτε στην εγκατάσταση τους ως ανεξάρτητη υποδομή σε φυσικούς εξυπηρετητές. Δεδομένου του ότι όλες οι προς εξέταση

λύσεις στηρίζονται τεχνολογικά σε Kubernetes, το οποίο δύναται να εγκατασταθεί είτε σε φυσικούς εξυπηρετητές είτε σε υποδομές IaaS, μπορεί με ασφάλεια να προκύψει το συμπέρασμα ότι όλες οι παραπάνω λύσεις είναι συμβατές με την υπάρχουσα υποδομή. Αναφορικά με την ευκολία ενσωμάτωσης, η λύση της πλατφόρμας OKD παρουσιάζεται ως η πιο συμβατή, αφού απαιτεί, μόνο, την εκτέλεση ενός κατάλληλα παραμετροποιημένου αρχείου εντολών (script).

4.3.3.4.3 Δυνατότητες αναβάθμισης

Η αναβάθμιση των τεχνολογικών λύσεων είναι κρίσιμος παράγοντας για να διασφαλιστεί η συνεχής καινοτομία, η ασφάλεια και η βελτίωση της απόδοσης των υποδομών πληροφορικής. Καθώς οι τεχνολογίες εξελίσσονται συνεχώς, η υποστήριξη αναβαθμίσεων με εύκολο και αποτελεσματικό τρόπο είναι ζωτικής σημασίας για την ασφάλεια και την διατήρηση των υποδομών και των υπηρεσιών που φιλοξενούνται στον οργανισμό. Αυτή η αξιολόγηση περιλαμβάνει την εξέταση της υποστήριξης που παρέχεται για τη διαδικασία αναβάθμισης.

Το OKD έχει επαρκείς δυνατότητες αναβάθμισης. Οι αναβαθμίσεις μπορούν να γίνουν μέσα από το γραφικό περιβάλλον ή μέσω της γραμμής εντολών (CLI). Οι διαδικασίες αναβάθμισης είναι διαμορφωμένες ώστε να μειώνουν τις διακοπές των υπηρεσιών, αλλά απαιτούν καλές γνώσεις σχετικά με την πλατφόρμα σε περίπτωση που οι αυτοματοποιημένες διαδικασίες αναβάθμισης αποτύχουν.

Η αναβάθμιση του Rancher εξαρτάται από τον τρόπο που αυτό έχει υλοποιηθεί (π.χ. Docker ή Kubernetes κ.α.) και συνήθως περιλαμβάνει την αναβάθμιση των αποθετηρίων και στην συνέχεια την εγκατάσταση της νέας έκδοσης μέσω γραμμής εντολών. Οι αναβαθμίσεις συνήθως απαιτούν ελάχιστη παρεμβολή και διακοπή των υπηρεσιών που φιλοξενούνται. Το Harvester, σε περίπτωση που αξιοποιηθεί ως τμήμα της λύσης, ακολουθεί παρόμοιες πρακτικές αναβάθμισης.

Το Kubesphere υποστηρίζει αναβαθμίσεις, είτε μέσω της διεπαφής χρήστη είτε μέσω της γραμμής εντολών. Οι αναβαθμίσεις απαιτούν κάποια εμπειρία, αλλά τα βήματα είναι καλά τεκμηριωμένα.

Η αναβάθμιση του Portainer μπορεί να γίνει εύκολα μέσω της διεπαφής του χρήστη, όπου ο χρήστης απλά μεταφορτώνει και εγκαθιστά την νεότερη έκδοση. Η αναβάθμιση του Kubernetes, στο οποίο φιλοξενούνται οι υπηρεσίες, του οποίου η διαχείριση γίνεται από το Portainer, εξαρτάται από την συγκεκριμένη διανομή του

Kubernetes που χρησιμοποιείται και μπορεί να ποικίλει ως προς την πολυπλοκότητα.

Στην αναβάθμιση της διανομής του Kubernetes ακολουθείται μια διαδικασία που περιλαμβάνει την αναβάθμιση του master node και στη συνέχεια των worker nodes, ενώ παράλληλα πρέπει να ληφθούν υπόψη οι αλλαγές στη διαμόρφωση και τη συμβατότητα των APIs. Οι διανομές του Kubernetes συνήθως παρέχουν εργαλεία ή τεκμηρίωση για την αναβάθμιση της συστοιχίας, ενώ συνεχίζεται η λειτουργία των εφαρμογών. Ωστόσο, η πρακτική αυτή απαιτεί σημαντική τεχνική γνώση και προσοχή στις λεπτομέρειες για να εξασφαλιστεί η συνεχής λειτουργία του συστήματος.

Όλες οι λύσεις, εκτός της συνδυαστικής λύσης του Portainer με συστοιχίες Kubernetes διαθέτουν αρκετά σαφείς και καθορισμένες διαδικασίες αναβάθμισης. Όλες οι λύσεις απαιτούν γνώσεις για την επαναφορά του συστήματος στην περίπτωση που οι διαδικασίες αναβάθμισης αποτύχουν.

4.3.3.4.4 Διαθέσιμη τεκμηρίωση

Η τεκμηρίωση είναι ένας κρίσιμος παράγοντας για την εγκατάσταση, την συντήρηση και παραμετροποίηση μιας λύσης. Όλες οι λύσεις, διαθέτουν εκτεταμένη τεκμηρίωση στον επίσημο ιστότοπο τους, καθώς παρέχουν οδηγίες για την εγκατάσταση, την διαχείριση και την αναβάθμιση τους, με εμπειρισταωμένο και οργανωμένο τρόπο ανάλογα με το επιθυμητό σενάριο υλοποίησης τους. Παράλληλα, παρέχουν οδηγίες για την αντιμετώπιση προβλημάτων, οδηγούς καλών πρακτικών και ποικιλία άλλων θεμάτων για προχωρημένους χρήστες και διαχειριστές.

Οι πιο διαδεδομένες διανομές Kubernetes, που πιθανά θα απαιτηθεί να εγκατασταθούν παράλληλα με τις προαναφερθείσες πλατφόρμες, διαθέτουν εκτεταμένη τεκμηρίωση και οδηγούς που διευκολύνουν τους διαχειριστές στην ομαλή ενσωμάτωση τους.

4.3.3.4.5 Δυνατότητα επίσημης υποστήριξης

Η δυνατότητα επίσημης υποστήριξης αποτελεί κρίσιμο παράγοντα για πολλές επιχειρήσεις και οργανισμούς που επιλέγουν να επενδύσουν σε τεχνολογίες ανοιχτού κώδικα. Πιο συγκεκριμένα, η επίσημη υποστήριξη παρέχει την ασφάλεια ότι τυχόν προβλήματα ή ερωτήματα που ανακύπτουν μπορούν να αντιμετωπιστούν

γρήγορα και αποτελεσματικά από εξειδικευμένους τεχνικούς. Όπως επίσης είναι σημαντική για την εξασφάλιση των συνεχών αναβαθμίσεων και βελτιώσεων που απαιτούνται.

Αναλυτικότερα, το OKD, που είναι η κοινοτική έκδοση του OpenShift, υποστηρίζεται από την Red Hat. Αν και η επίσημη υποστήριξη της Red Hat περιορίζεται στην εμπορική έκδοση του OpenShift. Υπάρχουν εταιρίες που παρέχουν εταιρική υποστήριξη και για το OKD. Το Rancher Labs, που αποτελεί μέρος της SUSE, παρέχει εμπορική υποστήριξη για το Rancher όπως και για το Harvester. Το Kubesphere παρέχει επίσημη υποστήριξη χωρίς να αναφέρεται αναλυτικά στα είδη υποστήριξης που μπορεί να παρέχει. Το Portainer παρέχει επίσημη υποστήριξη, αλλά προορίζεται για τη Επαγγελματική Έκδοση (Business Edition - BE) και όχι για την Κοινοτική Έκδοση (Community Edition - CE). Παρόλα αυτά υπάρχει δυνατότητα αναβάθμισης της Κοινοτικής Έκδοσης σε Επαγγελματική.

Το Kubernetes δεν παρέχει επίσημη υποστήριξη σε ιδιωτικές υλοποιήσεις. Παρόλα αυτά, πολλές εταιρίες, όπως η Google, η Microsoft, η IBM και η Red Hat παρέχουν εμπορική υποστήριξη για υλοποιήσεις Kubernetes. Το K3s είναι μια υλοποίηση της Rancher Labs, που ανήκει στην οικογένεια της SUSE από την οποία και υποστηρίζεται. Έτσι, μπορεί να προσφέρει επίσημη υποστήριξη για το K3s

4.3.3.4.6 Ωριμότητα Τεχνολογίας

Ένας από τους πιο κρίσιμους παράγοντες, που πρέπει να λαμβάνεται υπόψη κατά την επιλογή τεχνολογικών λύσεων για παραγωγικές υπηρεσίες, είναι το κατά πόσο αυτές είναι σταθερές και ώριμες. Συνήθως, η "πρώτη παραγωγική έκδοση" ενός λογισμικού αναφέρεται στην πρώτη έκδοση του λογισμικού που θεωρείται αρκετά σταθερή και ολοκληρωμένη για να χρησιμοποιηθεί σε παραγωγικό περιβάλλον. Αυτό σημαίνει ότι το λογισμικό έχει υποβληθεί σε επαρκείς δοκιμές, έχει επιλύσει σημαντικά ζητήματα ασφαλείας και απόδοσης, και παρέχει τις βασικές δυνατότητες και χαρακτηριστικά που περιγράφονται στην προδιαγραφή του προϊόντος. Σύμφωνα με τις ορθές πρακτικές στην τεχνολογία λογισμικού, η "πρώτη παραγωγική έκδοση" σημαίνεται με την έκδοση 1.0. Ωστόσο, η αρίθμηση των εκδόσεων μπορεί να διαφέρει ανάλογα με την εταιρεία ή τον οργανισμό που αναπτύσσει το λογισμικό. Τέλος, η επιλογή ώριμων λύσεων ελαχιστοποιεί

κινδύνους, όπως απροσδόκητες διακοπές λειτουργίας, προβλήματα συμβατότητας ή ελλείψεις δυνατότητας (features).

Οι λύσεις που έχουν προκριθεί αφορούν λογισμικά που διαθέτουν εκδόσεις σταθερές και με σήμανση του 1.0. Πιο συγκεκριμένα το OKD βρίσκεται στην έκδοση 4.12, το Kubernetes στην έκδοση 3.3.2, το Portainer CE στην έκδοση του 2.18.3, το Rancher στην έκδοση 2.7.4, το Harvester στην έκδοση 1.2, το Kubernetes στην έκδοση 1.27.2 και το K3s στην έκδοση 1.27.2+k3s1.

Ένας επιπλέον καίριος παράγοντας στην επιλογή κατάλληλων λύσεων αποτελεί η έκδοση των λογισμικών που είτε είναι προαπαιτούμενα για την εγκατάστασή τους, είτε αξιοποιούνται ως βιβλιοθήκες στον πηγαίο κώδικά τους. Μία λύση η οποία δεν χρησιμοποιεί σταθερές και αξιόπιστες εκδόσεις στα λογισμικά αυτά δεν μπορεί να θεωρηθεί ώριμη και σταθερή, ανεξάρτητα με την σήμανση που διαθέτει η έκδοση της.

4.3.3.4.7 Προαπαιτούμενη τεχνογνωσία

Η εγκατάσταση, η διαχείριση και η συντήρηση νεφούπολογιστικών υποδομών απαιτούν γνώσεις σχετικές με τις αρχές της υπολογιστικής νέφους, των δικτυακών συστημάτων, της διαχείρισης δεδομένων, της ασφάλειας πληροφοριών και γενικότερα της διαχείρισης υπολογιστικών πόρων. Η εγκατάσταση, η διαχείριση και η συντήρηση υποδομών φιλοξενίας και ενορχήστρωσης περιεκτών έχουν ως προαπαιτούμενο γνώσεις της τεχνολογίας των περιεκτών, των μηχανών εκτέλεσης τους όπως και γνώσεις σχετικές με τις μικροϋπηρεσίες και την αρχιτεκτονική εφαρμογών που βασίζονται σε αυτές.

Κατά συνέπεια, οι διαδικασίες μετασχηματισμού υποδομών για την φιλοξενία πολλαπλών νεφούπολογιστικών υπηρεσιών (IaaS, PaaS, SaaS, CaaS) μέσα από ένα ενοποιημένο ή διασυνδεδεμένο περιβάλλον προϋποθέτει όλη την προαναφερθείσα τεχνογνωσία χωρίς να εξαντλείται σε αυτή. Γνώσεις σχετικές με συστήματα Unix/Linux, γνώσεις προγραμματισμού, γνώσεις shell scripting και γνώσεις DevOps μπορούν να αποδειχθούν ιδιαίτερα χρήσιμες, ίσως και αναγκαίες για την πλήρη κατανόηση και αξιοποίηση των υποδομών αυτών. Παράλληλα, τεχνογνωσία και εμπειρία σε σχέση με την εφαρμογή καλών ή βέλτιστων πρακτικών και την επίλυση προβλημάτων θα διευκολύνει το τεχνικό προσωπικό στην ομαλότερη ενσωμάτωση των λύσεων αυτών.

Η ομάδα διαχείρισης συστημάτων και δικτύων του ΕΛΚΕΘΕ δύναται να θεωρηθεί ότι κατέχει την απαιτούμενη τεχνογνωσία αναφορικά με τις τεχνολογίες νεφούπολογιστικών υποδομών για την διάθεση υλισμικού ως υπηρεσία, όπως και ευρεία εμπειρία και γνώσεις σχετικά με δικτυακές υποδομές, διαχείριση υπολογιστικών πόρων, αλλά και εξειδικευμένες γνώσεις σχετικά με την διαχείριση συστημάτων Linux, στην παρακολούθηση δικτυακών υπηρεσιών και συστημάτων, καθώς και ικανοποιητικές γνώσεις ανάπτυξης λογισμικού και shell scripting. Ταυτόχρονα, διαθέτει επαρκή εμπειρία στην εγκατάσταση και υποστήριξη παραγωγικών υποδομών που πληρούν κριτήρια υψηλής διαθεσιμότητας, καταμερισμού φόρτου και αναστροφής αποτυχίας (fail over) όπως και συμφωνιών επιπέδου υπηρεσίας (SLAs).

Από την άλλη πλευρά, η εμπειρία και η τεχνογνωσία σε σχέση με τεχνολογίες περιεκτών, αλλά κυρίως με τεχνολογίες ενορχήστρωσης τους, είναι περιορισμένη, καθώς ο φορέας δεν διαθέτει κάποια πλατφόρμα ή υπηρεσία για την ενοποιημένη διαχείριση, φιλοξενία και διάθεση περιεκτών. Οι γνώσεις που διαθέτει το προσωπικό της ομάδας περιορίζονται σε ακαδημαϊκό επίπεδο και στην περιορισμένη εγκατάσταση και διαχείριση περιεκτών εντός μεμονωμένων εικονικών μηχανών.

Οι πλατφόρμες OKD, Rancher, Kubesphere, Portainer σε συνδυασμό (ή χωρίς) με διανομές Kubernetes που προκρίνονται προς εξέταση και εγκατάσταση, απαιτούν ένα σημαντικό επίπεδο τεχνογνωσίας, τεχνικών δεξιοτήτων και εμπειρίας, το οποίο δεν καλύπτεται στην ολότητα του από την ομάδα διαχείρισης συστημάτων και δικτύων, αλλά που κρίνεται ικανό και επαρκές για την υποστήριξη του επιθυμητού μετασχηματισμού.

4.3.4 Οικονομική Ανάλυση

Το κόστος υλοποίησης μιας τεχνολογικής λύσης αποτελεί κομβικό παράγοντα για την επιλογή της, αφού μία μη οικονομικά βιώσιμη επιλογή θα μπορούσε να οδηγήσει όχι μόνο σε αποτυχία εφαρμογής της λύσης αλλά και σε περεταίρω προβλήματα που αφορούν γενικότερα την επιχειρησιακή λειτουργία του οργανισμού, ακόμα και αν η τεχνολογική λύση έχει υλοποιηθεί επιτυχώς. Αυτό μπορεί να συμβεί διότι είτε δεν έχει γίνει σωστή αποτύπωση του κόστους απόκτησης της λύσης, είτε διότι το κόστος συντήρησης και υποστήριξης δεν μπορεί

να καλυφθεί από την απόδοση της επένδυσης. Ως συνέπεια αυτού ελλοχεύει ο κίνδυνος να καταστούν μη λειτουργικές υπηρεσίες κρίσιμες για τον οργανισμό, που υποστηρίζονται από την εκάστοτε λύση.

Ο τρόπος για να διασφαλίσει ο οργανισμός μια οικονομικά βιώσιμη λύση είναι η εκπόνηση μιας εμπειριστατωμένης μελέτης κόστους, η οποία θα περιλαμβάνει το κόστος απόκτησης λογισμικού, το κόστος απόκτησης υλικού, το κόστος συνδρομών και αδειών χρήσης, το κόστος εγκατάστασης και εκπαίδευσης όπως και το κόστος συντήρησης και υποστήριξης. Ακόμα, η μελέτη κόστους θα πρέπει να συνυπολογίζει τον χρονικό ορίζοντα χρήσης της εκάστοτε λύσης όπως και τις μελλοντικές απαιτήσεις για αναβαθμίσεις ή ενσωμάτωση νέων δυνατοτήτων. Τέλος, η τιμολογιακή πολιτική που ακολουθεί ο πάροχος της λύσης θα πρέπει να συμβαδίζει με τις πολιτικές προμήθειας που εφαρμόζει ο οργανισμός.

Όπως αναφέρθηκε κατά την αξιολόγηση της υφιστάμενης τεχνολογίας (βλ. 4.3.3.1), το ΕΛΚΕΘΕ αξιοποιεί στο μέγιστο δυνατό βαθμό προϊόντα και τεχνολογίες ανοιχτού κώδικα, ενώ παράλληλα επενδύει στο προσωπικό του για την εγκατάσταση και διαχείριση των εκάστοτε τεχνολογικών λύσεων. Η πρακτική αυτή ακολουθείται λόγω της ακαδημαϊκής ιδιότητας του φορέα αλλά και λόγω της χρηματοδοτικής πολιτικής που μπορεί να υποστηρίξει. Πιο αναλυτικά, οι ακαδημαϊκοί φορείς, βασικό μέλος των κοινοτήτων ανοιχτού κώδικα, παραμένουν σταθερά αφοσιωμένοι και χρήστες και παραγωγοί λογισμικού ανοιχτού κώδικα κάτι που αποτελεί αναπόσπαστο μέρος της κουλτούρας τους. Ταυτόχρονα, οι ακαδημαϊκοί φορείς, στην Ελλάδα, δεν διαθέτουν σταθερή χρηματοδότηση, ικανή να καλύψει όλες τις τεχνολογικές τους ανάγκες, οπότε και προτιμώνται λύσεις που είτε μπορούν να καλυφθούν από έκτακτα κονδύλια, είτε λύσεις που έχουν σταθερό και χαμηλό κόστος, όπως οι λύσεις ανοιχτού κώδικα.

Ως εκ τούτου οι λύσεις που προκρίθηκαν από την ομάδα διαχείρισης συστημάτων και δικτύων λαμβάνοντας υπόψιν τις λειτουργικές απαιτήσεις, τους τεχνολογικούς περιορισμούς και τις προτιμήσεις και τις δυνατότητες των χρηστών αφορούν προϊόντα ανοιχτού κώδικα με δυνατότητα υποστήριξης από επίσημους μεταπωλητές. Η επιλογή αυτή εξαλείφει το κόστος απόκτησης λογισμικού, ενώ καθιστά βιώσιμο το κόστος συντήρησης και υποστήριξης, συγκριτικά με ένα ιδιοπαγές προϊόν. Επιπρόσθετα, οι λύσεις αυτές δύναται να εγκατασταθούν με τη μορφή υπηρεσίας στην υπάρχουσα νεφούπολογιστική υποδομή εικονικών

μηχανών. Η δυνατότητα αυτή προσφέρει στον φορέα την ευκαιρία να υλοποιήσει και να δοκιμάσει λύσεις χωρίς να απαιτείται εξ' αρχής επένδυση σε υλικούς πόρους (ελάχιστο CAPEX), αλλά προσφέροντας την όποια επιθυμητή δυνατότητα επέκτασης της νέας υποδομής, με την προμήθεια απαιτούμενου υλικού ανάλογα και με τις ευκαιρίες χρηματοδότησης. Το κόστος εγκατάστασης και εκπαίδευσης είναι εξαρτώμενο από την υπάρχουσα τεχνογνωσία της ομάδας διαχείρισης συστημάτων και δικτύων και το οποίο σύμφωνα με τις διαθέσιμες προσφορές των επίσημων μεταπωλητών διατηρείται σε απόλυτα ανεκτά επίπεδα συγκριτικά με την υλοποίηση των επιλεγμένων λύσεων από εξωτερικούς συνεργάτες.

Συνοψίζοντας όλα τα παραπάνω προκύπτει ασφαλώς το συμπέρασμα πως όλες οι προς εξέταση λύσεις μπορούν να κριθούν ως βιώσιμες οικονομικά, αφού απαιτούν ελάχιστο κόστος αρχικής εγκατάστασης (CAPEX) και εύρος επιλογών σχετικά με τα λειτουργικά κόστη (OPEX). Παράλληλα, ακολουθούν τις πολιτικές του φορέα διασφαλίζοντας είτε τη βραχυπρόθεσμή, είτε τη μακροπρόθεσμη αξιοποίηση τους, προσφέροντας την απαραίτητη ευελιξία που αναζητεί ένας ακαδημαϊκός φορέας.

4.3.5 Επιλογή συμβατών λύσεων

Λαμβάνοντας υπόψιν όλη την προαναφερθείσα μελέτη και εστιάζοντας στην αξιολόγηση της συμβατότητας, όλες οι λύσεις που προκρίθηκαν (βλ. 4.3.3.2) δύναται να θεωρηθούν συμβατές σύμφωνα με την ανάλυση απαιτήσεων (βλ. 4.3.1). Η ομάδα διαχείρισης συστημάτων και δικτύων με σκοπό να αποκτήσει πλήρη εικόνα των λύσεων αυτών και των δυνατοτήτων τους επέλεξε να τις υλοποιήσει στις υπάρχουσες υποδομές του φορέα, ώστε να τις δοκιμάσει σε πραγματικές συνθήκες και να εξετάσει τις δυνατότητες ενσωμάτωσής τους. Οι προκριθείσες λύσεις προς εφαρμογή και υλοποίηση συνίστανται στις ακόλουθες:

- Rancher συνδυαστικά με Harvester, με διάφορες διανομές Kubernetes
- OKD αξιοποιώντας την δυνατότητα του για ενσωμάτωση στο oVirt

Όπως προαναφέρθηκε (βλ. 4.3.3.3), οι λύσεις αυτές προκρίνονται λόγω της διαλειτουργικότητάς τους με IaaS υποδομές, ενώ σε περίπτωση που οι δύο αυτές λύσεις σταθούν ανεπαρκείς ή καταστεί ανεπιτυχής η υλοποίησή τους για τον οποιονδήποτε λόγο, θα δοκιμαστούν επικουρικά και οι λύσεις Portainer

(συμπληρωματικά με διάφορες διανομές Kubernetes) και Kubesphere (με ή χωρίς διάφορες διανομές Kubernetes).

4.4 Υλοποίηση και παρουσίαση λύσεων

Η ομάδα διαχείρισης συστημάτων και δικτύων για την υλοποίηση των λύσεων επέλεξε να αξιοποιήσει την υπάρχουσα υποδομή του φορέα. Η επιλογή αυτή έγινε για να διευκολυνθεί η εργασία της ομάδας, δεδομένου ότι η εγκατάσταση νέας υποδομής αποτελεί χρονοβόρο και επιβαρυντικό παράγοντα, καθώς παράλληλα απαιτεί και επιπρόσθετο φυσικό εξοπλισμό. Πιο συγκεκριμένα, η εγκατάσταση και η δοκιμή τουλάχιστον δύο διαφορετικών τεχνολογικών λύσεων προϋποθέτει πολλαπλές εγκαταστάσεις εξυπηρετητών, νέες δικτυακές φυσικές υποδομές (switches, cabling) και υπηρεσίες (DNS, DHCP, NTP) όπως και τη χρήση επιπλέον συστημάτων αποθήκευσης. Η εκπλήρωση των προϋποθέσεων αυτών απαιτεί, με τη σειρά της, πληθώρα εργατωρών και νέο εξοπλισμό (χρόνος και κόστος). Δεδομένου ότι η ομάδα διαχείρισης συστημάτων και δικτύων πρέπει να αξιολογήσει τουλάχιστον δύο τεχνολογικές λύσεις και να επιλέξει μία εξ αυτών, κατέληξε οι αρχικές εγκαταστάσεις και δοκιμές να υλοποιηθούν στην υπάρχουσα IaaS υποδομή, ώστε η διαδικασία να είναι πιο αποδοτική, ευέλικτη και οικονομικά βιώσιμη. Εάν στη συνέχεια κριθεί απαραίτητο η ομάδα μπορεί να προχωρήσει σε περαιτέρω ενέργειες για προμήθεια και εγκατάσταση νέου εξοπλισμού. Ταυτόχρονα, η επιτυχής εγκατάσταση σε υπάρχουσα υποδομή εξασφαλίζει την συμβατότητα με την υποδομή αυτή και επιτρέπει την επέκταση / εγκατάσταση της λύσης, είτε με την επέκταση της διαθέσιμης IaaS υποδομής, είτε με την προσθήκη επιπλέον φυσικού εξοπλισμού.

Για την αξιολόγηση και την επιλογή της καταλληλότερης λύσης, η ομάδα διαχείρισης συστημάτων και δικτύων καθόρισε μια σειρά από κριτήρια, βάσει των οποίων, μετά από την βαθμονόμηση και βαθμολόγηση τους, θα προκύψει η τελική λύση που θα εγκατασταθεί παραγωγικά. Αυτά τα κριτήρια περιλαμβάνουν

- την ευκολία εγκατάστασης και συμβατότητας με την υπάρχουσα υποδομή IaaS
- την φιλικότητα διεπαφή χρήστη (UI)
- την επιτυχή εγκατάσταση δοκιμαστικών εφαρμογών

- τις δυνατότητες διαχείρισης των χρηστών και των πόρων (quotas)
- την απομόνωση του δικτύου
- τις δυνατότητες αναβάθμισης και επεκτασιμότητας της υποδομής
- τις δυνατότητες για υψηλή διαθεσιμότητα και επαναφορά της συστοιχίας μετά από πλήρη τερματισμό της λειτουργίας της.

4.4.1 Προετοιμασία υποδομής και Παραδοχές

Για την επίτευξη των δοκιμών εντός της υπάρχουσας IaaS υποδομής και εντός καθορισμένου χρονικού πλαισίου απαιτήθηκε να γίνουν κάποιες παραδοχές όπως και να παραμετροποιηθεί κατάλληλα η υπάρχουσα υποδομή. Αναλυτικότερα, υλοποιήθηκαν οι κατάλληλες τεχνικές αλλαγές ώστε να υποστηρίζεται το Nested Virtualization και το PCI Passthrough (SR-IOV). Οι δύο αυτές τεχνολογίες είναι προαπαιτούμενες για την δημιουργία εξυπηρετητών στην IaaS υποδομή, οι οποίοι θα μπορούν να φιλοξενούν εικονικές μηχανές. Επιπλέον η υποδομή IaaS (oVirt) αναβαθμίστηκε στην κατάλληλη έκδοση ώστε να μπορεί να υποστηρίξει τις πλέον σύγχρονες εκδόσεις του OKD, σύμφωνα με τον πίνακα εκδόσεων [69]. Παράλληλα, εισήχθησαν οι απαραίτητες εγγραφές στους DNS εξυπηρετητές, ενσωματώθηκαν στους DHCP εξυπηρετητές διαφορετικά υποδίκτυα για τους master και worker κόμβους των νέων υποδομών και παραμετροποιήθηκε ο δικτυακός εξοπλισμός για την υποστήριξη των αντίστοιχων απομονωμένων εικονικών δικτύων (VLANs). Η απαιτούμενη χωρητικότητα σε αποθηκευτικό χώρο καλύφθηκε με την προσθήκη μονάδων αποθήκευσης στην IaaS υποδομή, από τις υπάρχουσες διαθέσιμες SAN και NAS συστοιχίες.

Τέλος, επιλέχθηκε η διανομή K3s ως η διανομή που θα χρησιμοποιηθεί συνδυαστικά με τις λύσεις που στηρίζονται σε διανομές Kubernetes είτε για την εγκατάστασή τους, είτε για την διάθεση υπηρεσιών βασιζόμενη σε περιέκτες. Το K3s έχει μικρότερο αποτύπωμα από άλλες διανομές K8s, κάτι που το καθιστά ιδανικό για περιβάλλοντα με περιορισμένους πόρους, ενώ παράλληλα παραμένει συμβατό με τα πλέον χρησιμοποιούμενα πρότυπα Kubernetes, επιτρέποντας την χρήση όλων των κύριων λειτουργιών Kubernetes. Επιπλέον, διαθέτει απλοποιημένη διαδικασία εγκατάστασης και διαχείρισης καθιστώντας το πιο εύχρηστο και ευέλικτο για νέους χρήστες ή δοκιμαστικές εργασίες.

4.4.2 Rancher, με Harvester με διάφορες διανομές Kubernetes

Το Rancher είναι μια ανοιχτού κώδικα πλατφόρμα που διευκολύνει τη διαχείριση και τη λειτουργία των Kubernetes συστοιχιών. Σχεδιάστηκε για να ελέγχει και να διαχειρίζεται Deployments σε περιβάλλοντα Kubernetes ανεξάρτητα της υποδομής στην οποία υλοποιούνται. Το Rancher μπορεί να χρησιμοποιηθεί για τη διαχείριση των Kubernetes συστοιχιών που εκτελούνται σε διάφορα περιβάλλοντα, συμπεριλαμβανομένων εγκαταστάσεων εντός του οργανισμού (on-premise) - οργανισμοί που διαθέτουν τα δικά τους κέντρα δεδομένων, υπηρεσιών νέφους - διαχείριση των συστοιχιών που εκτελούνται σε δημόσιους παρόχους υπηρεσιών νέφους, όπως το Amazon Web Services (AWS), το Google Cloud Platform (GCP), το Microsoft Azure και άλλα, περιοχές άκρης (edge locations) - διαχείριση των συστοιχιών που βρίσκονται σε edge locations, όπως τα καταστήματα λιανικής πώλησης ή τα τοπικά γραφεία.

Με την χρήση του Rancher, μπορεί να επιτευχθεί η διαχείριση πολλαπλών συστοιχιών Kubernetes από μια ενιαία διεπαφή, η αναβάθμιση, η επέκταση και η διαγραφή συστοιχιών - ανεξάρτητα από το πού βρίσκονται - η παραμετροποίηση και η εφαρμογή πολιτικών ασφαλείας, όπως και η εγκατάσταση εφαρμογών με τη χρήση ελεύθερων (Helm Charts) ή ιδιωτικών αποθετηρίων. Μία καίριας σημασίας δυνατότητα του Rancher αποτελεί και η αυτοματοποιημένη δημιουργία νέων συστοιχιών Kubernetes, αφού διαθέτει έτοιμες διεπαφές για τους περισσότερους παρόχους υπηρεσιών νέφους, για την πλατφόρμα Harvester, καθώς και τη δυνατότητα δημιουργίας custom διεπαφών για άλλες IaaS υποδομές. Όπως προαναφέρθηκε (βλ. 4.3.3.2), το Harvester αποτελεί μια ανοιχτού κώδικα HCI πλατφόρμα που αξιοποιεί την τεχνολογία του Kubernetes για την παροχή υπολογιστικής ισχύος, αποθήκευσης και δικτύωσης σε μία ενιαία πλατφόρμα. Ο συνδυασμός του Rancher με το Harvester αποτελεί μία πολύ ενδιαφέρουσα λύση για τη διαχείριση περιβαλλόντων που βασίζονται στο Kubernetes, διότι προσφέρει τη δυνατότητα δημιουργίας και διαχείρισης εικονικών μηχανών καθώς και την ενσωμάτωση διαχείρισης αποθηκευτικών χώρων και δικτύωσης. Αναλυτικότερα, το Harvester προσφέρει ισχυρές δυνατότητες διαχείρισης εικονικών μηχανών, συμπεριλαμβανομένης της δυνατότητας δημιουργίας, τροποποίησης, διαγραφής και μετακίνησης (migration) εικονικών μηχανών μεταξύ διαφορετικών κόμβων. Επιπρόσθετα, παρέχει ενσωματωμένες λύσεις για αποθήκευση και δικτύωση,

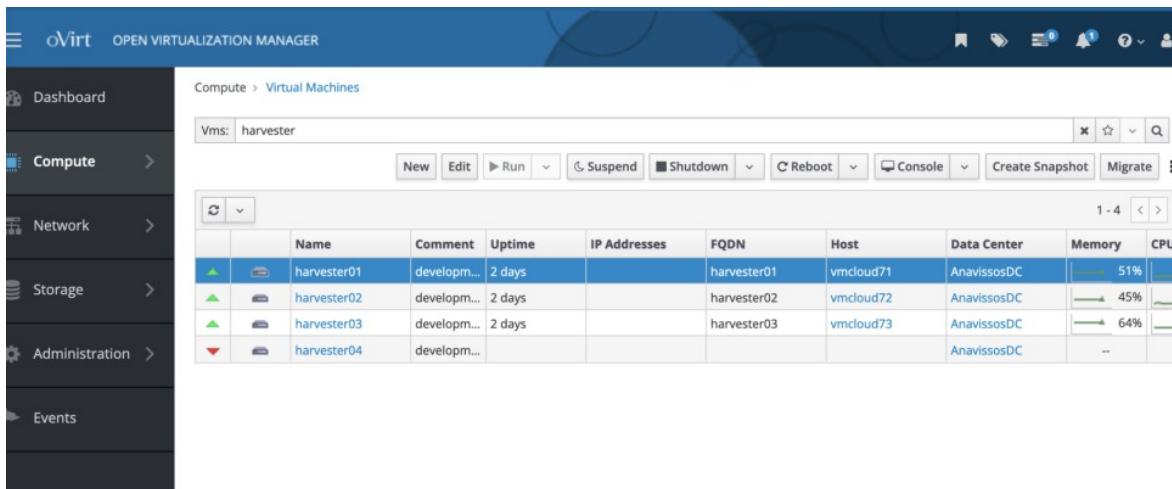
όπως το Longhorn για την αποθήκευση και το Mutlus για τη δικτύωση, που συνδυαστικά με το KubeVirt επιτρέπει στους χρήστες να δημιουργούν και να διαχειρίζονται εικονικές μηχανές και δίκτυα με τον ίδιο τρόπο που θα διαχειρίζονταν περιέκτες.

4.4.2.1 Εγκατάσταση υποδομής

Οι δύο τεχνολογίες, το Rancher και το Harvester, επιλέχθηκε να δοκιμαστούν συνδυαστικά, δεδομένου ότι παρουσιάζονται, από τον επίσημο πάροχο λογισμικού SUSE, ως μία ενιαία λύση. Η διαδικασία εγκατάστασης της υποδομής χωρίστηκε σε δύο στάδια, το στάδιο εγκατάστασης της υποδομής του Harvester και το στάδιο εγκατάστασης της πλατφόρμας Rancher, εντός της υποδομής του Harvester.

Εγκατάσταση του Harvester

Για την εγκατάσταση της υποδομής του Harvester δημιουργήθηκαν τρεις εικονικές μηχανές – η ελάχιστη απαίτηση για την πλήρη αξιοποίηση των λειτουργιών του Harvester, όπως η υψηλή διαθεσιμότητα, είναι τρεις κόμβοι - στην υποδομή IaaS (oVirt) στις οποίες παραχωρήθηκαν οι απαραίτητοι πόροι (αποθηκευτικά μέσα, μνήμη, επεξεργαστές, κάρτες δικτύου) ώστε να παραχθεί μία συστοιχία από κόμβους Harvester. Εκχωρήθηκαν μέσω τις IaaS υποδομής τα δίκτυα (VLANs) που δημιουργήθηκαν κατά την προετοιμασία της υποδομής, για το σκοπό αυτό, καθώς επίσης καταχωρήθηκαν και σχετικές DNS εγγραφές. Η διαδικασία εγκατάστασης του Harvester μπορεί να χαρακτηριστεί ομαλή, καθώς η τεκμηρίωση περιγράφει επαρκώς τα βήματα που πρέπει να ακολουθήσει ο διαχειριστής. Παρόλα αυτά απαιτήθηκε η εγκατάσταση να επαναληφθεί δεύτερη φορά, αφού η τελευταία stable έκδοση 1.1.1 που εγκαταστάθηκε αρχικά παρουσίαζε αρκετά προβλήματα στη διαχείριση των εικονικών μηχανών, όπως η αποτυχία μετακίνησης, σε πραγματικό χρόνο, εικονικών μηχανών και η δημιουργία κλώνων. Λόγω των προβλημάτων αυτών η ομάδα διαχείρισης συστημάτων και δικτύων επέλεξε να εγκαταστήσει την πλέον σταθερή έκδοση, που την συγκεκριμένη χρονική περίοδο ήταν η 1.0.1.



Στιγμιότυπο 1. Εικονικές μηχανές υποδομής Harvester



Στιγμιότυπο 2. Οθόνη επιτυχούς φόρτωσης λογισμικού Harvester

Εγκατάσταση Rancher

Για την εγκατάσταση του Rancher παραγωγικά απαιτείται η δημιουργία μιας συστοιχίας Kubernetes, ενώ για δοκιμαστική εγκατάσταση το Rancher μπορεί να υλοποιηθεί και σε περιέκτη Docker. Η δημιουργία της συστοιχίας Kubernetes υλοποιήθηκε σε εικονική μηχανή (AlmaLinux 9) που φιλοξενήθηκε στην υποδομή του Harvester. Σύμφωνα με τις παραδοχές που είχε θέσει η ομάδα η διανομή Kubernetes επιλέχθηκε ήταν η K3s και η δοκιμή πραγματοποιήθηκε σε single node συστοιχία.

```
[root@rancher01 ~]# cat /etc/*release
AlmaLinux release 9.1 (Line Lynx)
NAME="AlmaLinux"
VERSION="9.1 (Line Lynx)"
ID="almalinux"
ID_LIKE="rhel centos fedora"
VERSION_ID="9.1"
PLATFORM_ID="platform:el9"
PRETTY_NAME="AlmaLinux 9.1 (Line Lynx)"
ANSI_COLOR="0;34"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:almalinux:almalinux:9::baseos"
HOME_URL="https://almalinux.org/"
DOCUMENTATION_URL="https://wiki.almalinux.org/"
BUG_REPORT_URL="https://bugs.almalinux.org/"

ALMALINUX_MANTISBT_PROJECT="AlmaLinux-9"
ALMALINUX_MANTISBT_PROJECT_VERSION="9.1"
REDHAT_SUPPORT_PRODUCT="AlmaLinux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.1"
AlmaLinux release 9.1 (Line Lynx)
AlmaLinux release 9.1 (Line Lynx)
[root@rancher01 ~]#
[root@rancher01 ~]#
[root@rancher01 ~]#
[root@rancher01 ~]# kubectl version --short
Flag --short has been deprecated, and will be removed in the future. The --short output will become the default.
Client Version: v1.24.10+k3s1
Kustomize Version: v4.5.4
Server Version: v1.24.10+k3s1
[root@rancher01 ~]#
```

Στιγμιότυπο 3. Εικονική μηχανή για την εγκατάσταση του Rancher

Με την αξιοποίηση του εργαλείου διαχείρισης πακέτων HELM, την προσθήκη του αποθετηρίου rancher-stable και την συνδρομή της επίσημης τεκμηρίωσης πραγματοποιήθηκε επιτυχώς η εγκατάσταση του Rancher.

```
curl -sL https://get.k3s.io | INSTALL_K3S_VERSION=v1.24.10+k3s1
INSTALL_K3S_SKIP_SELINUX_RPM=true sh -s - server

helm repo add rancher-stable https://releases.rancher.com/server-charts/stable

kubectl create namespace cattle-system

kubectl apply -f https://github.com/cert-manager/cert-
manager/releases/download/v1.7.1/cert-manager.crds.yaml

helm repo add jetstack

helm repo update

export KUBECONFIG=/etc/rancher/k3s/k3s.yaml

helm install cert-manager jetstack/cert-manager --namespace cert-manager --
create-namespace --version v1.7.1

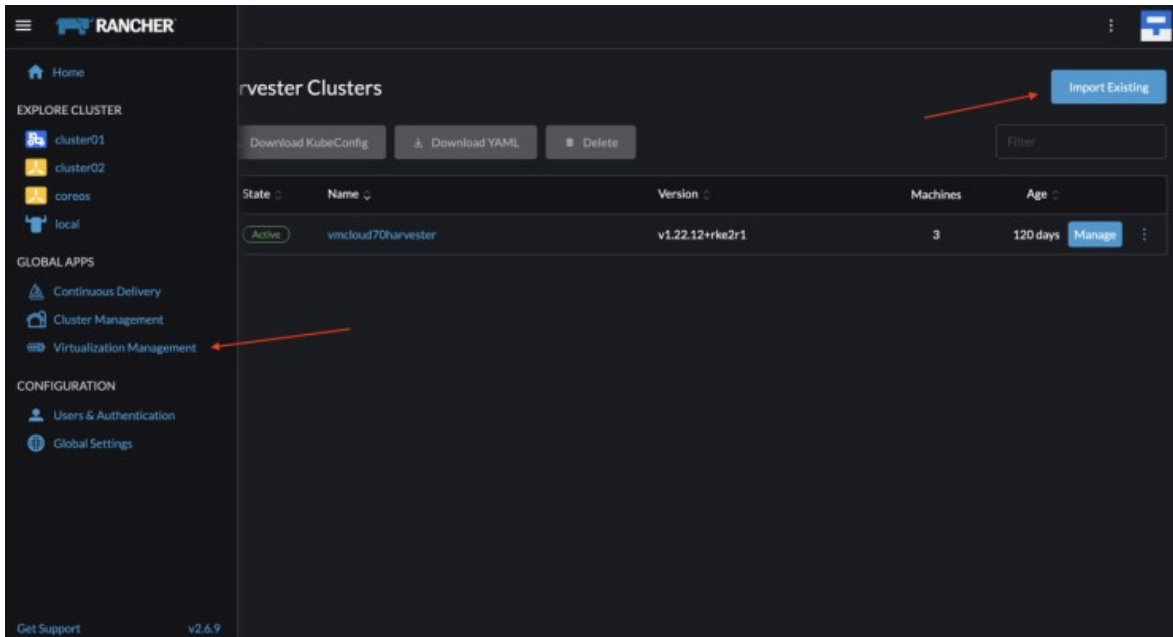
helm install rancher rancher-stable/rancher --namespace cattle-system --set
hostname=rancher01.intranet.hcmr.gr --set bootstrapPassword=admin --version
2.6.9 --set ingress.tls.source=rancher

echo https://rancher01.intranet.hcmr.gr/dashboard/?setup=$(kubectl get secret -
-namespace cattle-system bootstrap-secret -o go-
template='{{.data.bootstrapPassword|base64decode}}')
```

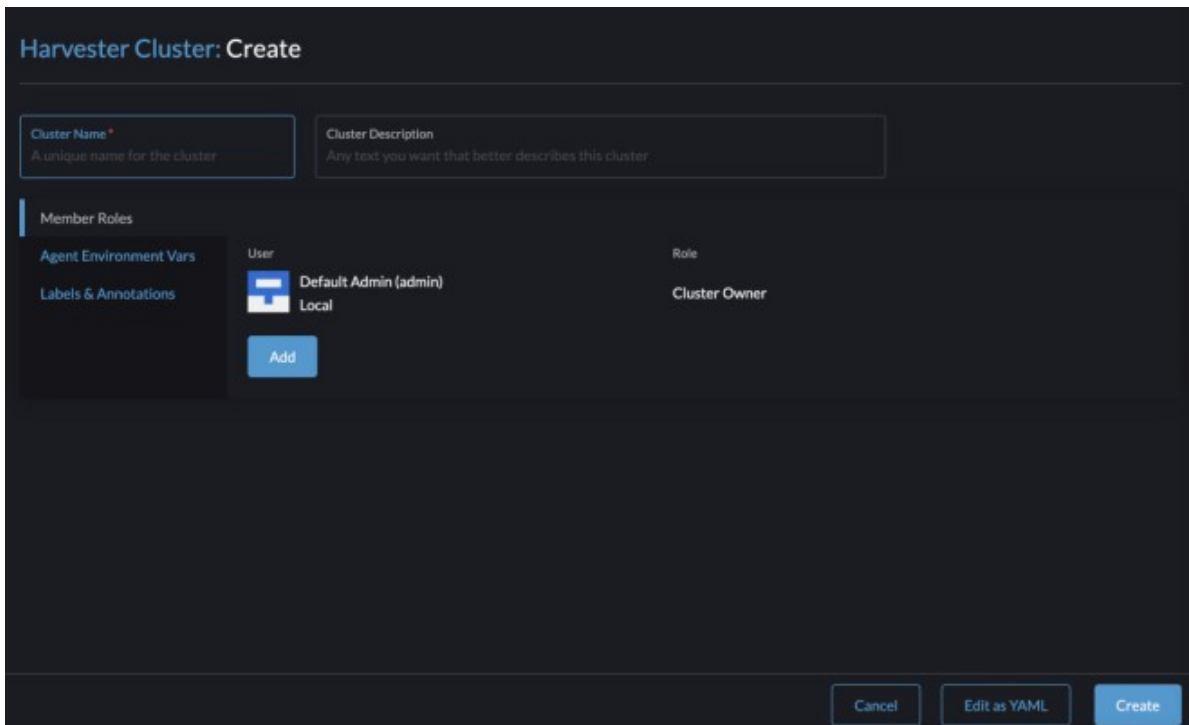
```
kubectl get secret --namespace cattle-system bootstrap-secret -o go-template='{{.data.bootstrapPassword|base64decode}}{{ "\n" }}'
```

Ενσωμάτωση Harvester στο Rancher

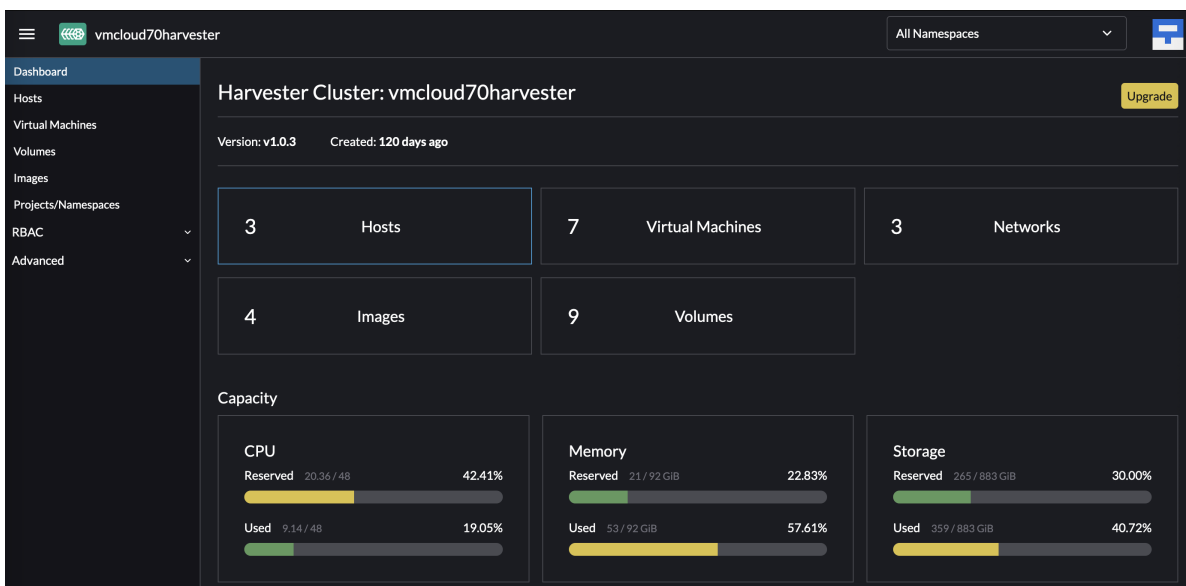
Η διαδικασία ενσωμάτωσης της IaaS υποδομής Harvester στο Rancher υλοποιήθηκε μέσα από το γραφικό περιβάλλον χρήστη που διαθέτει το Rancher, ακολουθώντας τη σχετική τεκμηρίωση.



Στιγμιότυπο 4. Ενσωμάτωση Harvester στο Rancher (1)



Στιγμιότυπο 5. Ενσωμάτωση Harvester στο Rancher (2)

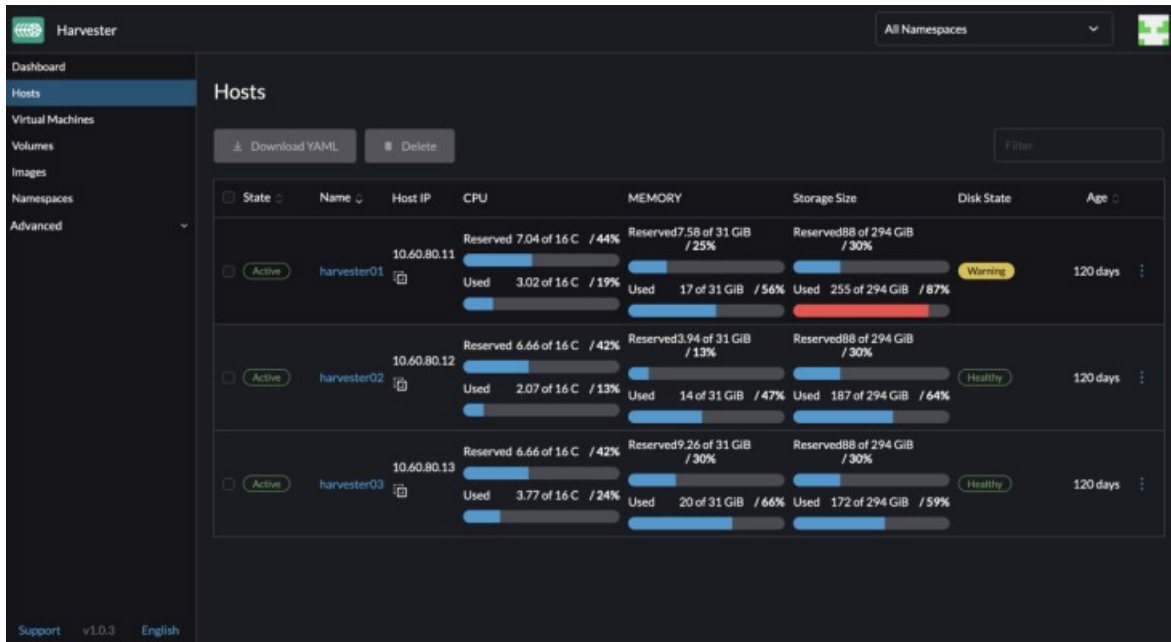


Στιγμιότυπο 6. Ολοκλήρωση διαδικασίας ενσωμάτωσης Harvester στο Rancher

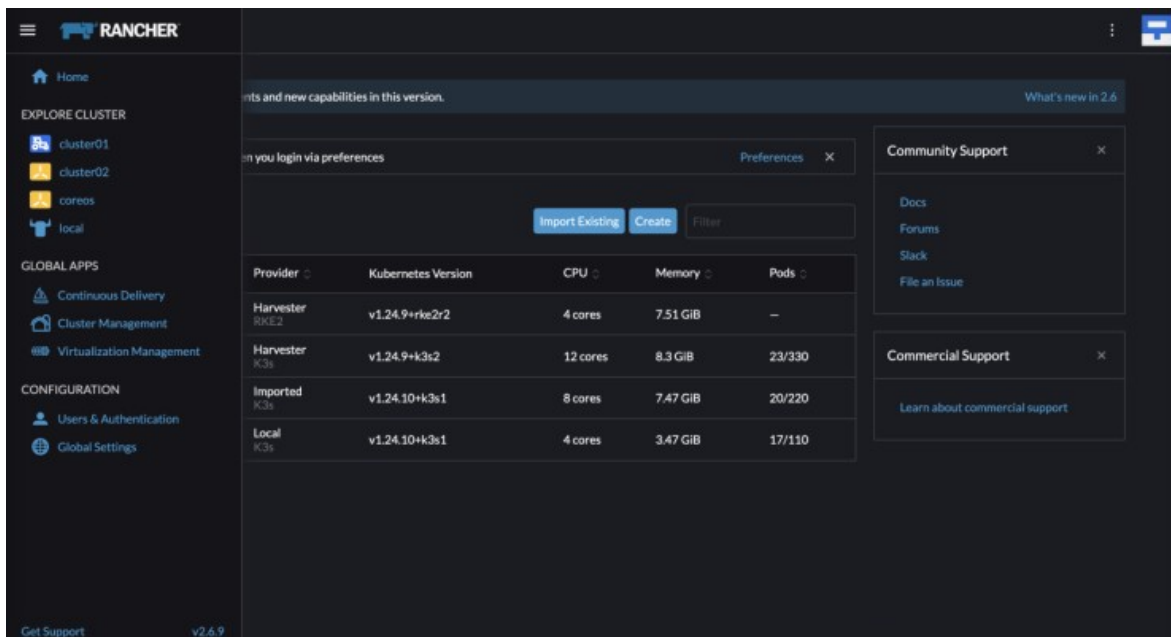
4.4.2.2 Προσβασιμότητα στο UI

Μετά την επιτυχή εγκατάσταση της υποδομής η ομάδα διαχείρισης συστημάτων και δικτύων περιηγήθηκε στις δικτυακές διεπαφές χρήστη του Harvester και του

Rancher ώστε να μελετήσει και να καταγράψει τις επιλογές και τις δυνατότητες που αυτές παρέχουν.



Στιγμιότυπο 7. Harvester: Γραφική διεπαφή

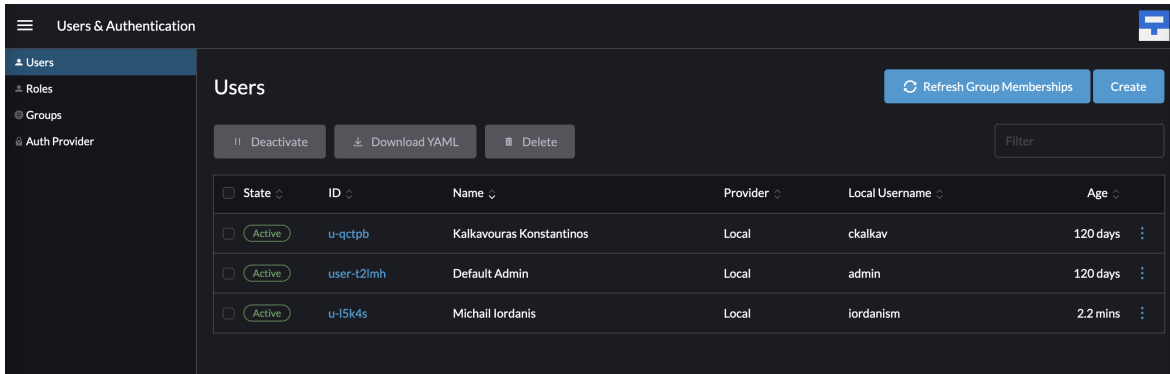


Στιγμιότυπο 8. Rancher: Γραφική διεπαφή

4.4.2.3 Δημιουργία χρηστών και διαχείριση πόρων

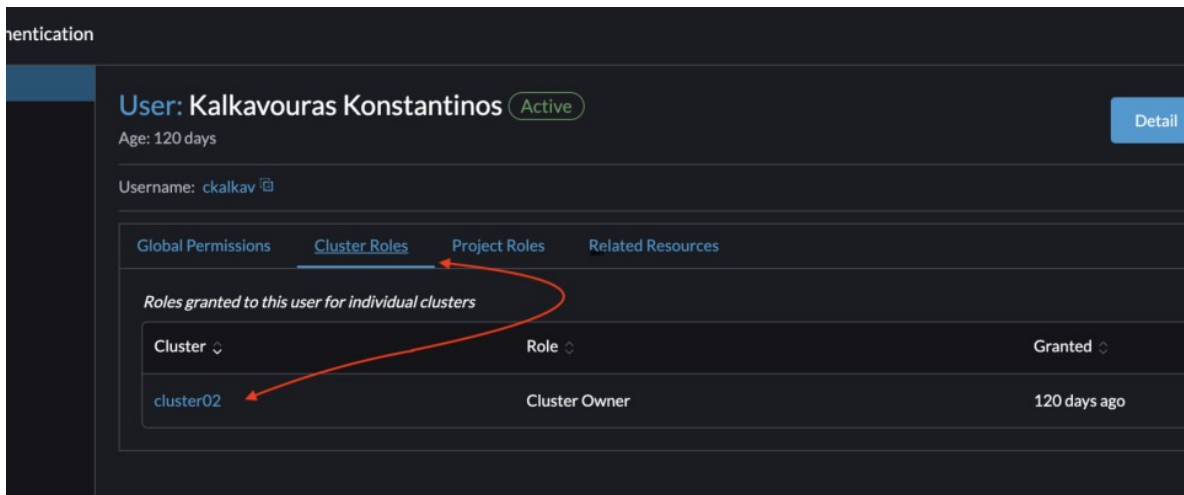
Η διαδικασία δημιουργίας χρηστών, στο Rancher, επιτυγχάνεται με απλό τρόπο μέσα από την πλοήγηση στις κατάλληλες διεπαφές για την διαχείριση χρηστών. Χρησιμοποιώντας είτε την τοπική βάση, είτε κάποιον εξωτερικό πάροχο

αυθεντικοποίησης όπως OpenLDAP που χρησιμοποιείται ευρέως στα ακαδημαϊκά ιδρύματα. Αντίστοιχα μέσα από τα menu τις διεπαφής δημιουργούνται ρόλοι και ομάδες χρηστών ώστε να επιτυγχάνεται αποδοτικά η πρόσβαση στους επιθυμητούς πόρους.



Στιγμιότυπο 9. Rancher: Διεπαφή χρηστών

Η προσέγγιση που ακολούθησε η ομάδα για τον διαμορισμό των πόρων και την αντιστοίχιση τους σε χρήστες, ή ομάδες χρηστών, στηρίχθηκε στον καθορισμό συστοιχιών Kubernetes που θα είναι προσβάσιμες από τους χρήστες ή τις ομάδες αυτές. Πρακτικά, οι διαχειριστές της υποδομής θα δημιουργούν συστοιχίες Kubernetes και θα τις αναθέτουν σε χρήστες ή ομάδες που θα έχουν αποκλειστικά πρόσβαση στις συστοιχίες αυτές. Με τον τρόπο αυτό οι διαχειριστές συστημάτων θα μπορούν να αξιοποιούν τις δυνατότητες αυτοματοποιημένης επέκτασης των συστοιχιών και γενικότερα της διαχείρισης της υποδομής, ενώ οι χρήστες πλήρη διαχείριση στις διανομές Kubernetes χωρίς να υπερβούν τους πόρους που έχουν χορηγηθεί. Με τον ίδιο τρόπο επιτυγχάνεται και η απομόνωση των δικτύων, αφού σε κάθε ομάδα εικονικών μηχανών που φιλοξενεί μια συστοιχία Kubernetes μπορεί να ανατεθεί διαφορετικό εικονικό δίκτυο (VLAN).

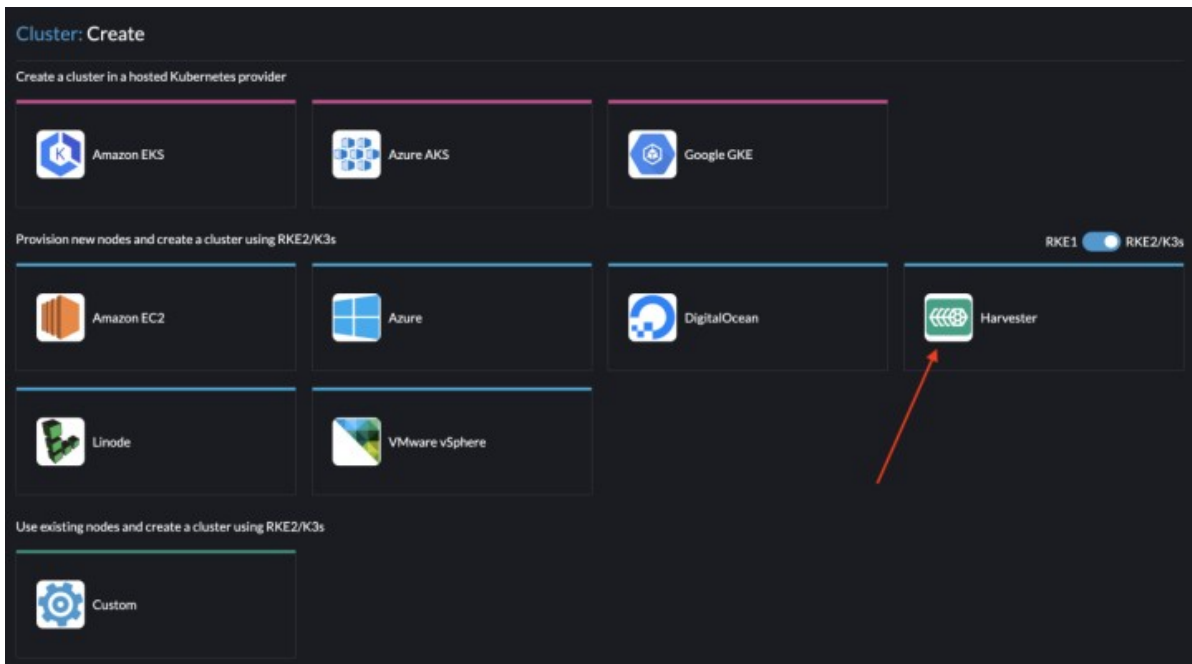


Στιγμιότυπο 10. Rancher: Διασύνδεση Kubernetes συστοιχίας με χρήση

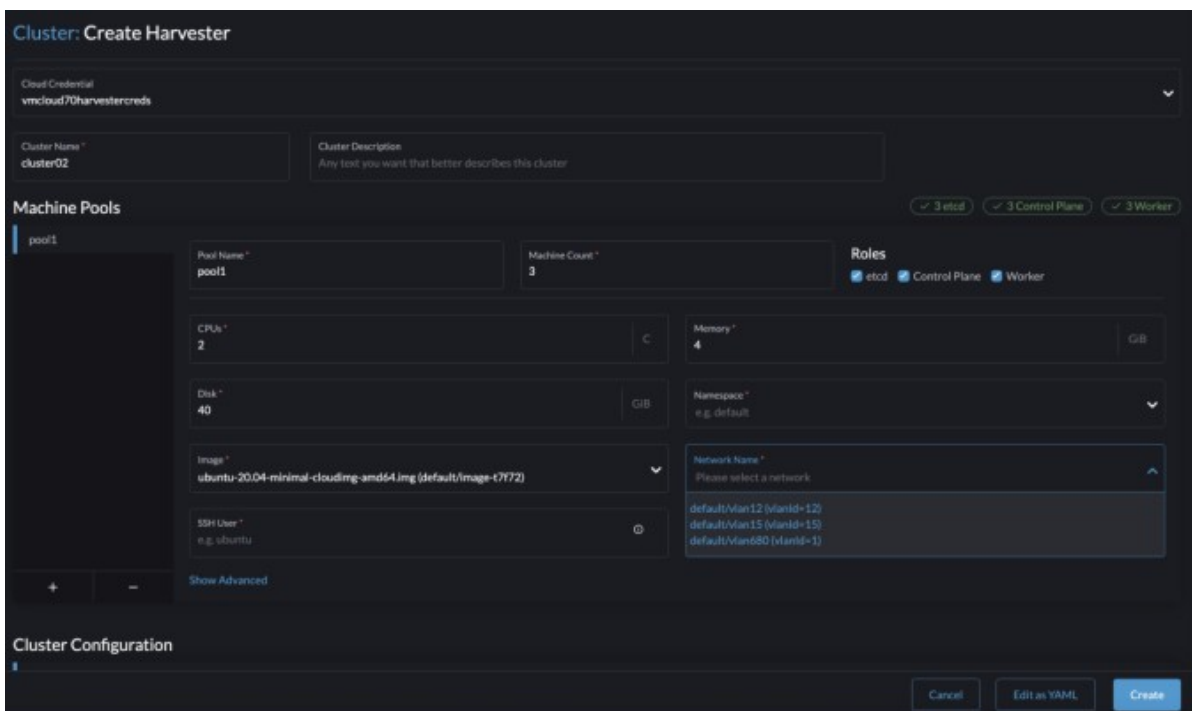
Για να υπάρχουν διαθέσιμες συστοιχίες Kubernetes οι οποίες θα ανατεθούν σε χρήστες, η ομάδα δημιούργησε δύο συστοιχίες (μία RKE και μία K3s) αξιοποιώντας την ενσωματωμένη υποδομή Harvester, ενώ ενσωμάτωσε και μία υπάρχουσα συστοιχία υλοποιημένη ανεξάρτητα στην IaaS υποδομή (oVirt). Αυτό επέτρεψε να ελεγχθεί και η διαλειτουργικότητα μεταξύ Rancher και Harvester και η συνδεσιμότητα της πλατφόρμας Rancher με τρίτες συστοιχίες Kubernetes, καθιστώντας τις δοκιμές όσο το δυνατόν πιο πλήρεις.

State	Name	Provider	Kubernetes Version	CPU	Memory	Pods
Active	cluster01	Harvester RKE2	v1.24.9+rke2r2	4 cores	7.51 GiB	36/220
Active	cluster02	Harvester K3s	v1.24.9+k3s2	12 cores	8.3 GiB	20/330
Active	coreos	Imported K3s	v1.24.10+k3s1	8 cores	7.47 GiB	20/220
Active	local	Local K3s	v1.24.10+k3s1	4 cores	3.47 GiB	17/110

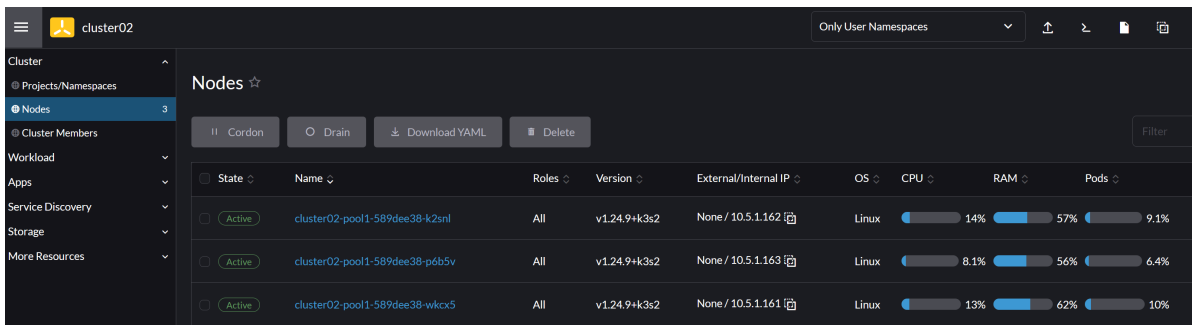
Στιγμιότυπο 11. Rancher: Λίστα Kubernetes συστοιχιών



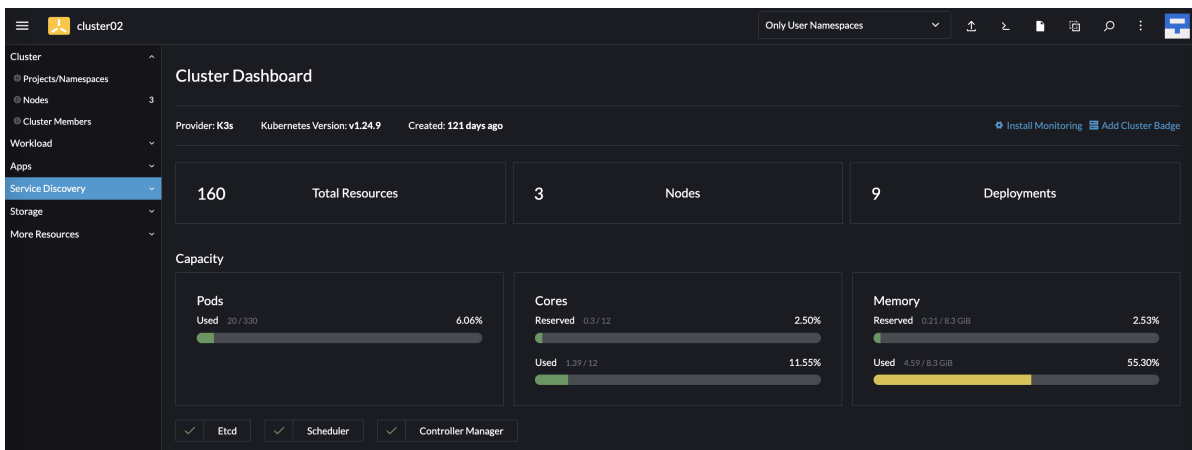
Στιγμιότυπο 12. Δημιουργία νέας Kubernetes συστοιχίας στο Harvester μέσω της διεπαφής του Rancher (1)



Στιγμιότυπο 13. Δημιουργία νέας Kubernetes συστοιχίας στο Harvester μέσω της διεπαφής του Rancher (2)



Στιγμιότυπο 14. Δημιουργία νέας Kubernetes συστοιχίας στο Harvester μέσω της διεπαφής του Rancher (3)



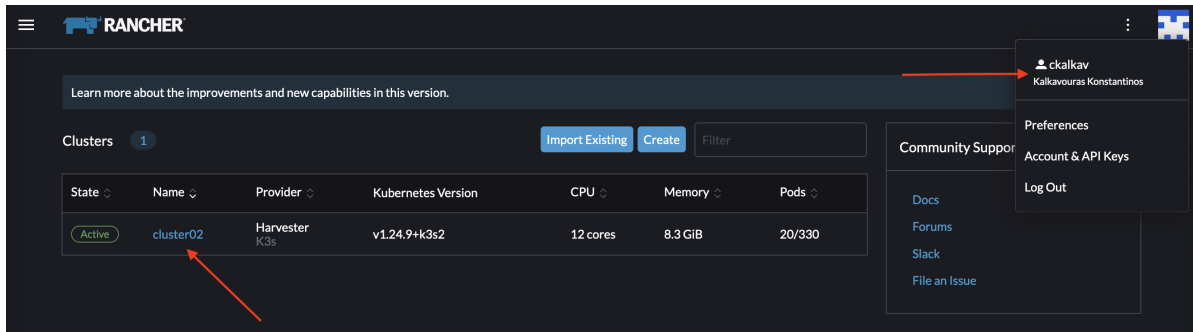
Στιγμιότυπο 15. Δημιουργία νέας Kubernetes συστοιχίας στο Harvester μέσω της διεπαφής του Rancher (4)

4.4.2.4 Εγκατάσταση δοκιμαστικής εφαρμογής

Για την δοκιμή εγκατάστασης εφαρμογών και υπηρεσιών η ομάδα επέλεξε να πραγματοποιήσει δύο διαφορετικούς τύπους εγκαταστάσεων. Ο πρώτος τρόπος αφορά την εγκατάσταση εφαρμογής μέσα από τα αποθετήρια εφαρμογών του Rancher που παρέχεται από το menu “Apps”. Ο δεύτερος τρόπος αφορά την εγκατάσταση custom Workload χρησιμοποιώντας εικόνες περιεκτών είτε από ιδιωτικά, είτε από δημόσια αποθετήρια.

Πρακτικά, ο πρώτος τρόπος επιβεβαιώνει την πιο αυτοματοποιημένη λειτουργία του Rancher για εγκατάσταση εφαρμογών, η οποία απαιτεί μικρότερη παρέμβαση του χρήστη και μπορεί να θεωρηθεί πιο φιλική σε σχέση με τον δεύτερο τρόπο, που απαιτεί μεγαλύτερη τεχνογνωσία από τον χρήστη και αναλυτικότερη παραμετροποίηση.

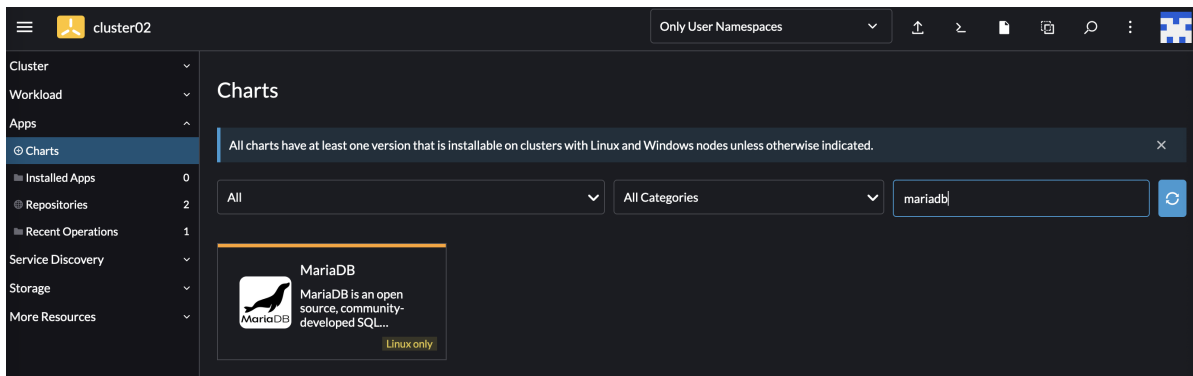
Οι δοκιμές υλοποιήθηκαν μέσω ενός τοπικού χρήστη στον οποίο είχε ανατεθεί μια συγκεκριμένη K3s συστοιχία, τριών κόμβων.



Στιγμιότυπο 16. Rancher: Επιλογή συστοιχίας για την εγκατάσταση δοκιμαστικών εφαρμογών

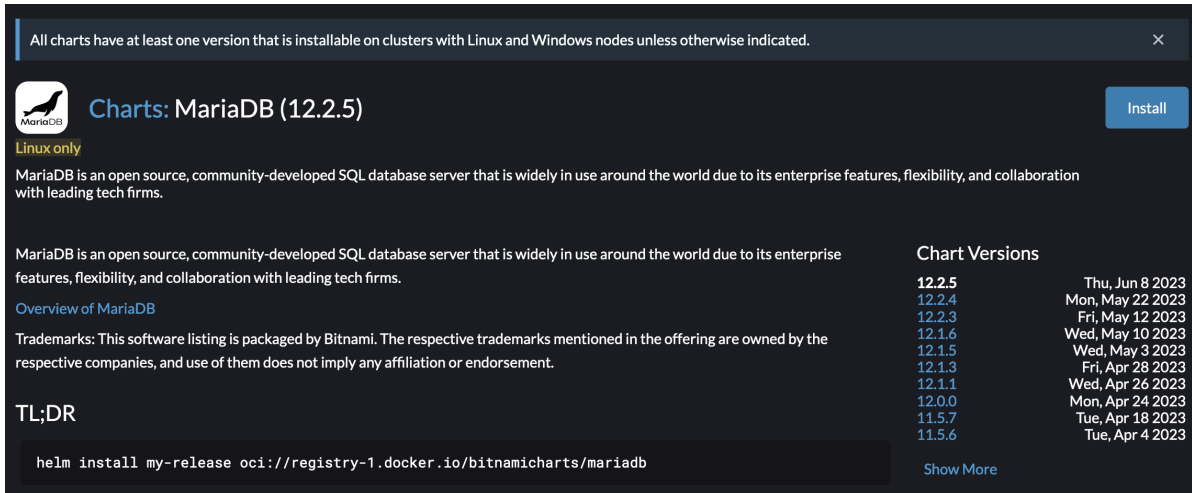
Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών ("Apps")

Για την εγκατάσταση της βάσης δεδομένων έγινε περιήγηση στην επιλογή Apps του γραφικού περιβάλλοντος, μέσω του οποίου αναζητήθηκε και επιλέχθηκε το αποθετήριο με την επιθυμητή έκδοση της MariaDB.



Στιγμιότυπο 17. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (1)

Παρατηρείται ότι για την εγκατάσταση της MariaDB χρησιμοποιείται ο διαχειριστής εφαρμογών (package manager) Helm, ο οποίος χρησιμοποιώντας το πρωτόκολλο OCI μεταφορτώνει την εικόνα περιέκτη από το δημόσιο αποθετήριο Docker Hub. Από τον σύνδεσμο είναι φανερό ότι χρησιμοποιείται η εικόνα περιέκτη με το όνομα MariaDB η οποία προσφέρεται από τον οργανισμό Bitnami.



All charts have at least one version that is installable on clusters with Linux and Windows nodes unless otherwise indicated.

Charts: MariaDB (12.2.5) Install

Linux only

MariaDB is an open source, community-developed SQL database server that is widely in use around the world due to its enterprise features, flexibility, and collaboration with leading tech firms.

MariaDB is an open source, community-developed SQL database server that is widely in use around the world due to its enterprise features, flexibility, and collaboration with leading tech firms.

[Overview of MariaDB](#)

Trademarks: This software listing is packaged by Bitnami. The respective trademarks mentioned in the offering are owned by the respective companies, and use of them does not imply any affiliation or endorsement.

TL;DR

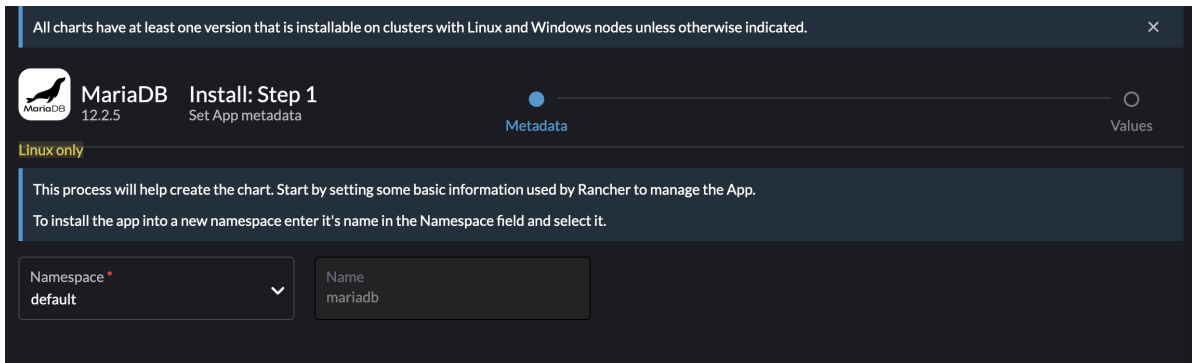
```
helm install my-release oci://registry-1.docker.io/bitnamicharts/mariadb
```

Chart Versions	Release Date
12.2.5	Thu, Jun 8 2023
12.2.4	Mon, May 22 2023
12.2.3	Fri, May 12 2023
12.1.6	Wed, May 10 2023
12.1.5	Wed, May 3 2023
12.1.3	Fri, Apr 28 2023
12.1.1	Wed, Apr 26 2023
12.0.0	Mon, Apr 24 2023
11.5.7	Tue, Apr 18 2023
11.5.6	Tue, Apr 4 2023

[Show More](#)

Στιγμιότυπο 18. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (2)

Στην συνέχεια επιλέγεται ένα συγκεκριμένο namespace με σκοπό τον διαχωρισμό και την απομόνωση των πόρων από άλλα projects και διατίθεται το yaml αρχείο της εφαρμογής προς επισκόπηση και πιθανή τροποποίηση, ώστε η εφαρμογή να παραμετροποιηθεί σύμφωνα με τις ανάγκες του χρήστη.



All charts have at least one version that is installable on clusters with Linux and Windows nodes unless otherwise indicated.

MariaDB 12.2.5 **Install: Step 1** Set App metadata

Progress: Metadata (filled), Values (empty)

Linux only

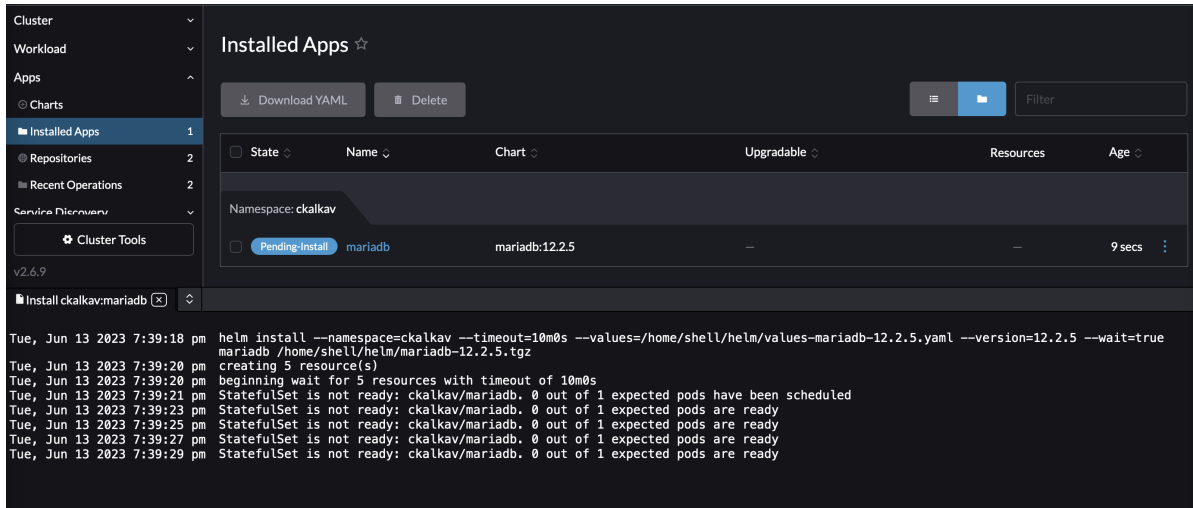
This process will help create the chart. Start by setting some basic information used by Rancher to manage the App.

To install the app into a new namespace enter it's name in the Namespace field and select it.

Namespace * Name

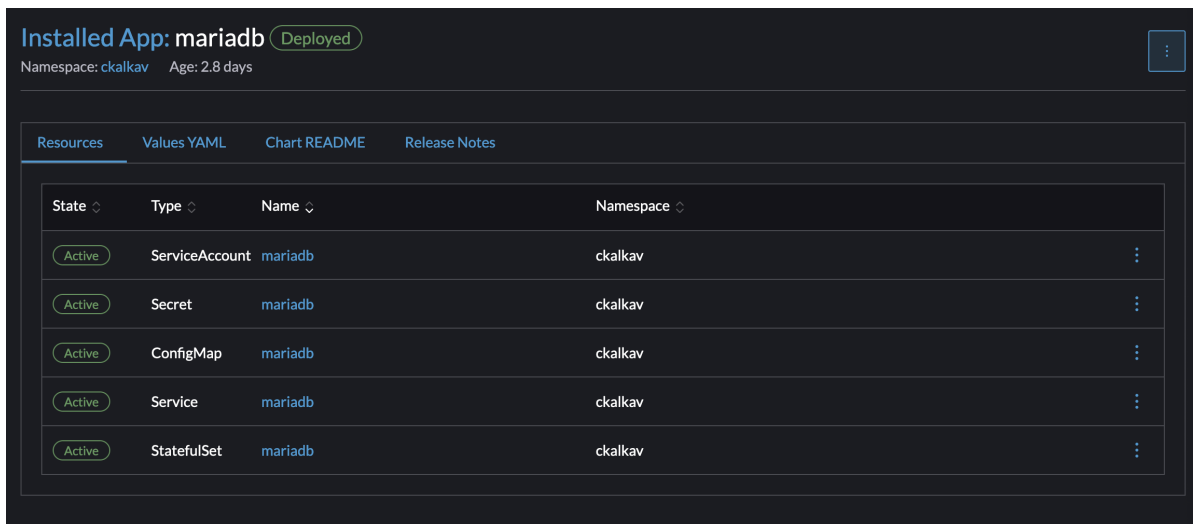
Στιγμιότυπο 19. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (3)

Το επόμενο βήμα αφορά την εγκατάσταση της εφαρμογής στην υποδομή, η οποία πραγματοποιείται σε διάστημα δευτερολέπτων.



Στιγμιότυπο 20. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (4)

Μετά την ολοκλήρωση της εγκατάστασης η εφαρμογή προβάλλεται στη λίστα εγκατεστημένων εφαρμογών – Apps – του χρήστη, από όπου μπορεί να διαχειριστεί περαιτέρω την εφαρμογή. Αναλυτικότερα, μέσα από το διαχειριστικό περιβάλλον υπάρχει δυνατότητα να διαχειριστεί όλους τους πόρους που χρησιμοποιεί η εφαρμογή, καθώς και να αξιοποιήσει το εργαλείο εντολών γραμμής kubectl για - τις ίδιες με το γραφικό περιβάλλον αλλά και - πιο εξειδικευμένες λειτουργίες.



Στιγμιότυπο 21. Rancher: Εγκατάσταση βάσης δεδομένων MariaDB μέσω διεπαφής εφαρμογών (5)

Η εφαρμογή MariaDB κατά την αυτοματοποιημένη υλοποίηση της δεν συνοδεύεται από κάποια ρύθμιση για εξωτερική πρόσβαση, ουσιαστικά προσφέροντας μόνο την εσωτερική Cluster IP. Αυτό συμβαίνει διότι στα πλαίσια

της αρχιτεκτονικής των μικροϋπηρεσιών τέτοιου είδους υπηρεσίες συνηθίζεται να είναι προσβάσιμες μέσω κάποιου ενδιάμεσου API ή τρίτης εφαρμογής. Για το λόγο αυτό η ομάδα διαχείρισης συστημάτων και δικτύων ενσωμάτωσε μια υπηρεσία εξισορρόπησης φόρτου (load balancer), ώστε η βάση δεδομένων να είναι προσβάσιμη από οποιοδήποτε κόμβο της συστοιχίας στην οποία φιλοξενείται.

Τέλος, η ομάδα συνδέθηκε στην βάση δεδομένων από εξωτερικό δίκτυο και εκτέλεσε βασικές εντολές για να επιβεβαιωθεί η ομαλή λειτουργία της υπηρεσίας.

```
[root@www-v03 ~]# ip -br a
lo                UNKNOWN          127.0.0.1/8  ::1/128
eth0              UP                195.251.37.36/26  fe80::21a:4aff:fe19:9d06/64
[root@www-v03 ~]# mysql -u root -h 10.5.1.163 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 48891
Server version: 10.11.4-MariaDB Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| my_database |
| mysql |
| performance_schema |
| sys |
| test |
+-----+
6 rows in set (0.00 sec)

MariaDB [(none)]> use test;
Database changed
MariaDB [test]> show tables;
Empty set (0.00 sec)

MariaDB [test]> CREATE TABLE Persons (
  ->   PersonID int,
  ->   LastName varchar(255),
  ->   FirstName varchar(255),
  ->   Address varchar(255),
  ->   City varchar(255)
  -> );
Query OK, 0 rows affected (0.04 sec)

MariaDB [test]> show tables;
+-----+
| Tables_in_test |
+-----+
| Persons |
+-----+
1 row in set (0.00 sec)

MariaDB [test]> quit;
Bye
[root@www-v03 ~]#
```

Στιγμιότυπο 22. Rancher: Έλεγχος λειτουργίας βάσης δεδομένων MariaDB

Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου

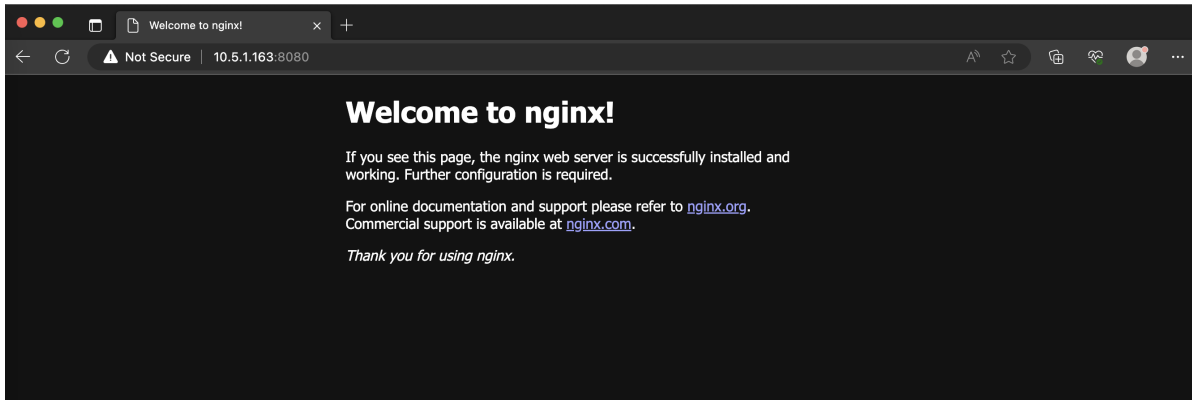
Για την εγκατάσταση του εξυπηρετητή δικτύου έγινε περιήγηση στην επιλογή Workload → Deployments → Create του γραφικού περιβάλλοντος. Στη διεπαφή του Create συμπληρώθηκαν βασικά χαρακτηριστικά της εφαρμογής, όπως το όνομα της, η εικόνα περιέκτη που θα χρησιμοποιηθεί και ορίστηκαν οι εξισορροπητές φόρτου για την προσβασιμότητα της από εξωτερικά δίκτυα.

Στιγμιότυπο 23. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (1)

Η εγκατάσταση της υπηρεσίας ολοκληρώθηκε επιτυχώς με την δημιουργία ενός Pod και δύο Services, ενώ η πρόσβαση στην υπηρεσία επιτυγχάνεται μέσω φυλλομετρητή από εσωτερικά ή εξωτερικά δίκτυα.

State	Name	Image	Ready	Up-to-date	Available	Restarts	Age	Health
Active	nginx	nginx	1/1	1	1	0	4.3 mins	

Στιγμιότυπο 24. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (2)



Στιγμιότυπο 25. Έλεγχος λειτουργίας εξυπηρετητή δικτύου (webserver) nginx

Με βάση τις παραπάνω επιτυχείς δοκιμές, η υποδομή Rancher, φαίνεται να προσφέρει την βασική λειτουργικότητα για την εγκατάσταση εφαρμογών περιεκτών με φιλικό και αποτελεσματικό τρόπο. Παράλληλα, η διεπαφή του χρήστη διαθέτει όλες τις απαραίτητες λειτουργίες, ώστε η διαχείριση των εφαρμογών να ολοκληρώνεται μέσα από αυτή, χωρίς να απαιτείται η συμμετοχή του διαχειριστή.

4.4.2.5 Επέκταση υποδομής

Η δυνατότητα επέκτασης μια νεφούπολογιστικής υποδομής αποτελεί έναν από τους πιο κρίσιμους παράγοντες για την αποδοτική και μακροχρόνια χρήση της. Η επέκταση της συγκεκριμένης υποδομής εξαρτάται τόσο από την επέκταση της IaaS υποδομής (Harvester), όσο και από την επέκταση των συστοιχιών Kubernetes που φιλοξενούν τις υπηρεσίες. Η διαχείριση των δυνατοτήτων επέκτασης, μέσα από το διαχειριστικό περιβάλλον Rancher, διευκολύνει τους διαχειριστές της υποδομής απεμπλέκοντας τους από χειροκίνητες διαδικασίες επιρρεπείς σε λάθη.

Αναφορικά με την επέκταση της υποδομής Harvester και σύμφωνα με την επίσημη τεκμηρίωση, η διαδικασία περιλαμβάνει την εγκατάσταση του κατάλληλου λειτουργικού συστήματος (έκδοση Harvester) στον φυσικό ή εικονικό εξυπηρετητή και την παραμετροποίηση του, ώστε να ενσωματωθεί στην υπάρχουσα συστοιχία. Προϋποθέσεις για την επιτυχή ενσωμάτωση, της επιπλέον υποδομής, αποτελεί η ύπαρξη των απαιτούμενων πόρων στον εξυπηρετητή και η ορθή παραμετροποίηση, η οποία εκτελείται μέσω της διεπαφής εγκατάστασης που διαθέτει το Harvester. Εκτελώντας την παραπάνω διαδικασία η ομάδα πρόσθεσε στην υπάρχουσα συστοιχία Harvester έναν ακόμα κόμβο πιστοποιώντας με αυτό τον τρόπο την τεκμηρίωση και τις δυνατότητες επέκτασης.

Compute > Virtual Machines

Vms: harvester

Name	Comment	Uptime	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics	Status	Description
harvester01	developm...	6 days		vmcloud71	AnavissosDC	59%	28%	--	VNC	Up	development
harvester02	developm...	6 days		vmcloud72	AnavissosDC	54%	25%	--	VNC	Up	
harvester03	developm...	6 days		vmcloud73	AnavissosDC	65%	29%	--	VNC	Up	
harvester04	developm...	1 min		vmcloud74	AnavissosDC	3%	9%	--	VNC	Up	

Στιγμιότυπο 26. Επέκταση υποδομής Harvester

Οι δυνατότητες επέκτασης των συστοιχιών Kubernetes που φιλοξενούνται στην IaaS υποδομή του Harvester περιορίζονται είτε στην οριζόντια επέκταση με την προσθήκη επιπλέον εικονικών μηχανών στην υπάρχουσα συστοιχία, είτε στην κάθετη επέκταση των ήδη χρησιμοποιούμενων εικονικών μηχανών. Στα πλαίσια δοκιμής των δυνατοτήτων αυτών η ομάδα τροποποίησε την επεξεργαστική ισχύ και την μνήμη των εικονικών μηχανών μιας συστοιχίας και στη συνέχεια πρόσθεσε δύο επιπλέον κόμβους στην συστοιχία αυτή. Η όλη διαδικασία μπορεί να υλοποιηθεί μέσω της γραφικής διεπαφής του Rancher παραμετροποιώντας κατάλληλα την δεξαμενή (pool) μηχανημάτων που απαρτίζει την συστοιχία. Στην ίδια λογική, το Rancher προσφέρει τη δυνατότητα προσθήκης επιπλέον δεξαμενής μηχανημάτων στην ίδια συστοιχία.

Cluster: Harvester - cluster02 (Active)
 Namespace: fleet-default Age: 124 days

Making changes to cluster configuration may result in nodes reprovisioning. For more information see the [documentation](#).

Cloud Credential: vmcloud70harvestercreds

Cluster Name: cluster02
 Cluster Description: Any text you want that better describes this cluster

Machine Pools: 5 etcd, 5 Control Plane, 5 Worker

pool1

Pool Name: pool1
 Machine Count: 5
 Roles: etcd, Control Plane, Worker

CPUs: 4
 Memory: 3 GIB
 Disk: 12 GIB
 Namespace: default
 Image: ubuntu-20.04-minimal-cloudimg-amd64.img (default/image-t7f72)
 Network Name: default/vlan15 (vlanId=15)
 SSH User: ubuntu

Buttons: Cancel, Edit as YAML, Save

Στιγμιότυπο 27. Rancher: Επέκταση Kubernetes συστοιχίας με προσθήκη επιπλέον κόμβων (1)

Virtual Machines Create

Stop Restart Start Download YAML Delete Filter

State	Name	CPU	Memory	IP Address	Node	Age
Namespace: default						
Running	almarancher01	4	4 Gi	10.60.80.51	harvester03	124 days
Off	almarancher02	4	4 Gi			124 days
Running	cluster01-pool1-4aa5f6...	2	4 Gi	10.5.1.154	harvester03	123 days
Running	cluster01-pool1-4aa5f6...	2	4 Gi	10.5.1.155	harvester01	123 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.162	harvester01	124 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.179	harvester01	2.5 mins
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.163	harvester02	124 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.161	harvester03	124 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.146	harvester02	2.5 mins

Στιγμιότυπο 28. Rancher: Επέκταση Kubernetes συστοιχίας με προσθήκη επιπλέον κόμβων (2)

Cluster: cluster02 Active

Namespace: fleet-default Age: 124 days Detail Config

Provisioner: K3s Machine Provider: Harvester Kubernetes Version: v1.24.9+k3s2 Machine Pools: 1 Machines: 5

[Machine Pools](#) [Provisioning Log](#) [Snapshots](#) [Conditions](#) [Related Resources](#)

Filter

State	Name	Node	IP Address	OS	Roles	Age
Pool: cluster02-pool1						
Unknown - No Location / No Size (cluster02-pool1-589dee38)						
Running	cluster02-pool1-5497b69946-bgc87	cluster02-pool1-589dee38-wkcx5	10.5.1.161	Linux	All	124 days
Running	cluster02-pool1-5497b69946-c97mt	cluster02-pool1-589dee38-p6b5v	10.5.1.163	Linux	All	124 days
Running	cluster02-pool1-5497b69946-n7qjh	cluster02-pool1-589dee38-zrcnn	10.5.1.146	Linux	All	12 mins
Running	cluster02-pool1-5497b69946-pb827	cluster02-pool1-589dee38-m7l5s	10.5.1.179	Linux	All	12 mins
Running	cluster02-pool1-5497b69946-qqr4c	cluster02-pool1-589dee38-k2snl	10.5.1.162	Linux	All	124 days

Στιγμιότυπο 29. Rancher: Επέκταση Kubernetes συστοιχίας με προσθήκη επιπλέον κόμβων (3)

State	Name	Roles	Version	External/Internal IP	OS	CPU	RAM	Pods	Age
Active	cluster02-pool1-589dee38-k2snl	All	v1.24.9+k3s2	None / 10.5.1.162	Linux	18%	62%	9.1%	124 days
Active	cluster02-pool1-589dee38-m7l5s	All	v1.24.9+k3s2	None / 10.5.1.179	Linux	7.6%	39%	5.5%	6 mins
Active	cluster02-pool1-589dee38-p6b5v	All	v1.24.9+k3s2	None / 10.5.1.163	Linux	11%	57%	5.5%	124 days
Active	cluster02-pool1-589dee38-wkcx5	All	v1.24.9+k3s2	None / 10.5.1.161	Linux	19%	63%	10%	124 days
Active	cluster02-pool1-589dee38-zrcnn	All	v1.24.9+k3s2	None / 10.5.1.146	Linux	7.9%	37%	5.5%	6 mins

Στιγμιότυπο 30. Rancher: Επέκταση Kubernetes συστοιχίας με προσθήκη επιπλέον κόμβων (4)

Συμπληρωματικά, με την επέκταση των υποδομών και των συστοιχιών μελετάται και η δυνατότητα επέκτασης της συστοιχίας που φιλοξένη την πλατφόρμα Rancher. Όπως προαναφέρθηκε (βλ. 4.4.4.1), η πλατφόρμα Rancher είναι εγκατεστημένη σε συστοιχία K3s, σε εικονικές μηχανές στην HCI υποδομή Harvester. Η επέκταση της συγκεκριμένης συστοιχίας μπορεί να υλοποιηθεί με την προσθήκη εικονικών μηχανών, στη συστοιχία αυτή, με μη αυτοματοποιημένο τρόπο. Πρακτικά, θα πρέπει να δημιουργηθούν επιπλέον εικονικές μηχανές και να ακολουθηθεί η διαδικασία προσθήκης νέων κόμβων στην συστοιχία K3s, σύμφωνα με την επίσημη τεκμηρίωση.

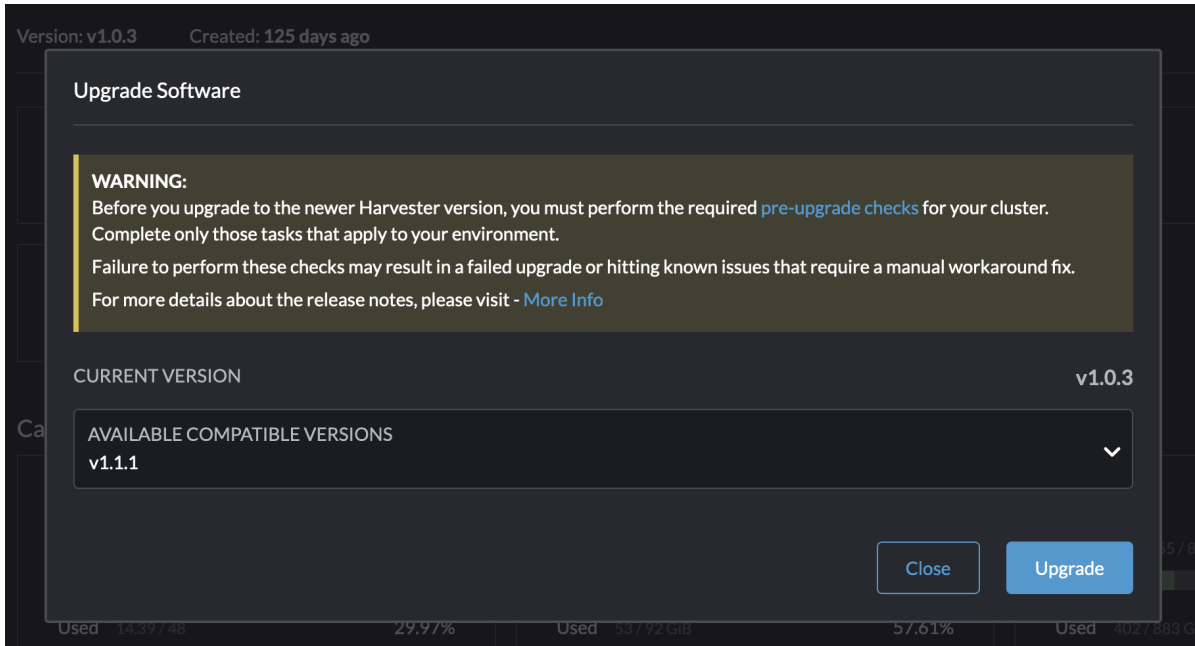
4.4.2.6 Αναβάθμιση υποδομής

Η συνολική αναβάθμιση της υποδομής απαιτεί την αναβάθμιση της IaaS υποδομής Harvester και την αναβάθμιση της πλατφόρμας Rancher. Οι αναβαθμίσεις θα πρέπει να υλοποιούνται με τρόπο τέτοιο που θα εξασφαλίζει τη συμβατότητα μεταξύ των διαθέσιμων εκδόσεων. Πιο συγκεκριμένα, πριν από κάθε αναβάθμιση ο διαχειριστής θα πρέπει να συμβουλευτεί την επίσημη τεκμηρίωση και τον πίνακα συμβατότητας εκδόσεων, δεδομένου ότι για την πλήρη λειτουργικότητα της υποδομής είναι απαραίτητη η διαλειτουργικότητα των δύο πλατφορμών. Επιπλέον, πρέπει να λαμβάνεται υπόψιν η συμβατότητα αναβάθμισης μεταξύ των εκδόσεων του ίδιου λογισμικού, αφού πολλές φορές δεν υποστηρίζονται οι απευθείας αναβαθμίσεις από την τρέχουσα στην τελευταία σταθερή έκδοση.

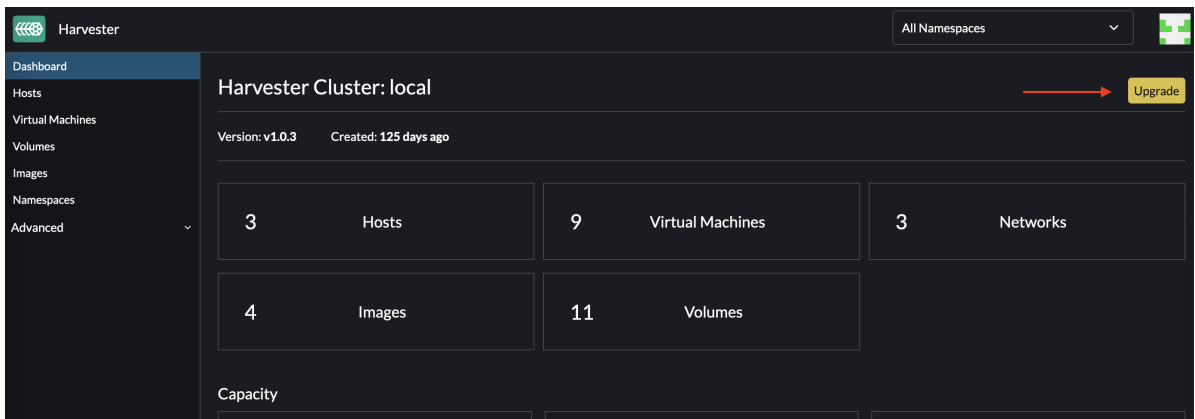
Η αναβάθμιση του Harvester σύμφωνα με την επίσημη τεκμηρίωση μπορεί να υλοποιηθεί με δύο τρόπους. Ο πρώτος τρόπος αφορά την αναβάθμιση μέσω της γραφικής διεπαφής, ενώ ο δεύτερος τρόπος που απευθύνεται κυρίως σε πιο απομονωμένα περιβάλλοντα αφορά την επανεγκατάσταση της νεότερης έκδοσης σε κάθε ένα από τους εξυπηρετητές στους οποίους φιλοξενείται. Η αναβάθμιση μέσω της γραφικής διεπαφής μπορεί σαφώς να χαρακτηριστεί φιλικότερη και εξασφαλίζει κατ' ελάχιστον την συμβατότητα μεταξύ των εκδόσεων της ίδιας πλατφόρμας, αλλά ενέχει μεγαλύτερης δυσκολίας διαχείριση σε περίπτωση αστοχίας.

Rancher Version	Harvester Version	Harvester Node Driver Supported K8S Versions
v2.7.3 - v2.7.4	v1.1-1.1.2	RKE1 & RKE2 v1.23, v1.24
v2.6.11	v1.0.3-1.1.2	RKE1 & RKE2 v1.23, v1.24
v2.6.9	v1.0.3-1.1.1	RKE1 & RKE2 v1.23, v1.24
v2.6.6 - v2.6.8	v1.0.2 - v1.0.3	RKE1 & RKE2 v1.22, v1.23, v1.24
v2.6.5	v1.0.1 - v1.0.2	RKE1 & RKE2 v1.22, v1.23, v1.24
v2.6.4	v1.0.1	RKE1 & RKE2 v1.22, v1.23, v1.24
v2.6.3	v1.0.0	RKE1 & RKE2 v1.22, v1.23, v1.24

Εικόνα 3. Πίνακας συμβατότητας εκδόσεων Rancher, Harvester και K8S



Στιγμιότυπο 31. Αναβάθμιση Harvester μέσω γραφικής διεπαφής χρήστη (1)



Στιγμιότυπο 32. Αναβάθμιση Harvester μέσω γραφικής διεπαφής χρήστη (2)

Η αναβάθμιση της πλατφόρμας Rancher συστήνεται, σύμφωνα με την επίσημη τεκμηρίωση, να αναβαθμιστεί αξιοποιώντας το εργαλείο διαχείρισης πακέτων Helm, εγκαθιστώντας το αποθετήριο με την επιθυμητή έκδοση και εκτελώντας τις κατάλληλες εντολές σε εργαλείο γραμμής εντολών. Για την εκτέλεση της διαδικασίας με ασφάλεια και τη δυνατότητα επαναφοράς σε περίπτωση αστοχίας, προτρέπεται σαφώς ο διαχειριστής να έχει λάβει αντίγραφο ασφαλείας της Kubernetes συστοιχίας που φιλοξενεί την πλατφόρμα του Rancher. Για την λήψη του αντιγράφου ασφαλείας παρέχεται λογισμικό (Rancher backups) το οποίο μπορεί να εγκατασταθεί και να εκτελεστεί μέσα από το γραφικό περιβάλλον διεπαφής χρήστη.

4.4.2.7 Έλεγχος για υψηλή διαθεσιμότητα

Για να εξασφαλιστεί η ομαλή και αξιόπιστη λειτουργία της υποδομής, είναι απαραίτητο να υλοποιηθούν δοκιμές που να επιβεβαιώνουν τις δυνατότητες παροχής υψηλής διαθεσιμότητας. Οι δοκιμές αυτές σχεδιάζονται για να εκτιμηθεί η ανθεκτικότητα, η αποδοτικότητα και η απόκριση της υποδομής σε καταστάσεις αποτυχίας ενός ή περισσότερων σημείων της υποδομής, που μπορεί να οφείλονται είτε στην αστοχία υλικού, είτε στον αυξημένο φόρτο εργασίας, είτε σε κακόβουλες επιθέσεις.

Η ομάδα διαχείρισης συστημάτων και δικτύων για να εξακριβώσει τις δυνατότητες υψηλής διαθεσιμότητας διαμόρφωσε συγκεκριμένα σενάρια δοκιμών με σκοπό την κάλυψη των συνηθέστερων συμβάντων. Πιο συγκεκριμένα, υλοποίησε δοκιμές που περιλάμβαναν αστοχία κόμβων και δικτύων. Η αστοχία κόμβων αφορά είτε κόμβους της συστοιχίας που απαρτίζουν την IaaS υποδομή, είτε κόμβους των συστοιχιών Kubernetes που φιλοξενούν τις υπηρεσίες / εφαρμογές. Κατά αντιστοιχία, η αστοχία στην δικτυακή υποδομή περιλαμβάνει αστοχία στις συνδέσεις μεταξύ των κόμβων ή συνολικότερα του εξοπλισμού.

Το πρώτο σενάριο, που αναπαράχθηκε, αφορούσε την αστοχία διακομιστών που απαρτίζουν την συστοιχία Harvester, η οποία πρακτικά αποτελεί το δομικό λίθο της υποδομής. Η συστοιχία της δοκιμής αποτελούταν από τρεις κόμβους και σύμφωνα με το σενάριο ένας κόμβος τέθηκε εκτός λειτουργίας, με την υποδομή να καταλήγει να προσφέρει υπηρεσίες από τους δύο εναπομείναντες κόμβους. Στη συνέχεια πραγματοποιήθηκε επαναφορά του εκτός λειτουργίας κόμβου και ελέγχθηκε η ομαλή ενσωμάτωση του στην υποδομή.

Name	Comment	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics	Status	Description
harvester01	development		vmcloud71	AnavissosDC	52%	20%	--	VNC	Up	development
harvester02	development		vmcloud72	AnavissosDC	47%	17%	--	VNC	Up	
harvester03	development		vmcloud73	AnavissosDC	66%	30%	--	VNC	Up	

Στιγμιότυπο 33. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (1)

Name	Comment	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics	Status	Description
harvester01	development			AnavissosDC	--	--	--	None	Down	development
harvester02	development		vmcloud72	AnavissosDC	47%	30%	--	VNC	Up	
harvester03	development		vmcloud73	AnavissosDC	66%	42%	--	VNC	Up	

Στιγμιότυπο 34. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (2)

Virtual Machines

Stop Restart Download YAML Delete Filter

State	Name	CPU	Memory	IP Address	Node	Age
Running	cluster01-pool1-4aa5f6...	2	4 Gi	10.5.1.155	harvester01	125 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.179	harvester01	1.7 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.146	harvester02	1.7 days
Running	almarancher01	4	4 Gi	10.60.80.51	harvester03	126 days
Running	cluster01-pool1-4aa5f6...	2	4 Gi	10.5.1.154	harvester03	125 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.161	harvester03	126 days

Στιγμιότυπο 35. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (3)

Virtual Machines

Stop Restart Download YAML Delete Filter

State	Name	CPU	Memory	IP Address	Node	Age
Not Ready	cluster02-pool1-589de...	4	3 Gi	10.5.1.179	harvester01	1.7 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.146	harvester02	1.7 days
Running	cluster01-pool1-4aa5f6...	2	4 Gi	10.5.1.154	harvester03	125 days
Not Ready	cluster01-pool1-4aa5f6...	2	4 Gi	10.5.1.155	harvester01	125 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.161	harvester03	126 days
Running	almarancher01	4	4 Gi	10.60.80.51	harvester03	126 days

Στιγμιότυπο 36. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (4)

Virtual Machines

Stop Restart Download YAML Delete Filter

State	Name	CPU	Memory	IP Address	Node	Age
Starting	cluster02-pool1-589de...	4	3 Gi			1.7 days
Guest VM is not reported as running						
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.146	harvester02	1.7 days
Running	cluster01-pool1-4aa5f6...	2	4 Gi	10.5.1.154	harvester03	125 days
Starting	cluster01-pool1-4aa5f6...	2	4 Gi			125 days
Guest VM is not reported as running						
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.161	harvester03	126 days
Running	almarancher01	4	4 Gi	10.60.80.51	harvester03	126 days

Στιγμιότυπο 37. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (5)

State	Name	CPU	Memory	IP Address	Node	Age
Running	cluster01-pool1-4aa5f6...	2	4 Gi	10.5.1.155	harvester02	125 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.146	harvester02	1.7 days
Running	almarancher01	4	4 Gi	10.60.80.51	harvester03	126 days
Running	cluster01-pool1-4aa5f6...	2	4 Gi	10.5.1.154	harvester03	125 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.179	harvester03	1.7 days
Running	cluster02-pool1-589de...	4	3 Gi	10.5.1.161	harvester03	126 days

Στιγμιότυπο 38. Δοκιμή υψηλής διαθεσιμότητας υποδομής Harvester (6)

Η δοκιμή στέφθηκε με επιτυχία δεδομένου ότι καμία υπηρεσία που φιλοξενούσε η υποδομή δεν τέθηκε εκτός λειτουργίας, παρά μόνο οι εικονικές μηχανές που φιλοξενούνταν στους διακομιστές που απενεργοποιήθηκαν και επανεκκινήθηκαν στον διαθέσιμο εναπομείναντα διακομιστή, σε χρονικό διάστημα μικρότερο του λεπτού. Το σενάριο αυτό προκάλεσε υποβάθμιση της συστοιχίας, κάτι όμως που ήταν αναμενόμενο, αφού συστοιχία κάτω των τριών κόμβων, δεν διαθέτει ανοχή σφαλμάτων στην βάση δεδομένων etcd και πρακτικά αξιοποιεί μόνο ένα κόμβο, καθώς υπάρχει απαίτηση για μονό πλήθος κόμβων. Αντιθέτως οι master και worker κόμβοι παρέχουν ανοχή σφαλμάτων όταν το πλήθος τους είναι κατ' ελάχιστον δύο. Αντίστοιχη δοκιμή εκτελέστηκε και με την απενεργοποίηση ζεύξεων δικτύου σε διακομιστές Harvester, κάτι που έγινε αντιληπτό στη συστοιχία ως αστοχία κόμβου.

Η δεύτερη δοκιμή προσομοίωσε την αστοχία εικονικών μηχανών που συνέθεταν συστοιχίες Kubernetes για την φιλοξενία υπηρεσιών και εφαρμογών. Πιο αναλυτικά, κάθε συστοιχία Kubernetes στηρίζεται σε έναν πλήθος εικονικών μηχανών που έχουν δημιουργηθεί με αυτόματες ή χειροκίνητες διαδικασίες στην IaaS υποδομή που παρέχει το Harvester. Οι εικονικές αυτές μηχανές που αποτελούν τους master ή / και worker κόμβους των συστοιχιών Kubernetes είναι πιθανό να τεθούν εκτός λειτουργίας είτε λόγω του κόμβου στον οποίο φιλοξενούνται, είτε λόγω άλλων παραγόντων όπως σφάλματα κώδικα (bugs), αστοχίες δικτύου κ.α.. Κατά αντιστοιχία του πρώτου σεναρίου, απενεργοποιήθηκαν εικονικές μηχανές μιας επιλεγμένης συστοιχίας είτε με την πλήρη απενεργοποίηση τους, είτε με την διακοπή της δικτυακής τους σύνδεσης. Η συστοιχία Kubernetes

ανταποκρίθηκε στις προδιαγραφές της διατηρώντας ενεργές τις υπηρεσίες και τις εφαρμογές που παρείχε, καθιστώντας την δοκιμή επιτυχή.

State	Name	Version	Provider	Machines	Age
Updating	cluster01	v1.24.9+rke2r2	Harvester RKE2	0 of 2	125 days
Updating	cluster02	v1.24.9+k3s2	Harvester K3s	2 of 3	126 days
Active	coreos	v1.24.10+k3s1	Imported K3s	2	108 days
Active	local	v1.24.10+k3s1	Local K3s	1	126 days

Στιγμιότυπο 39. Δοκιμή υψηλής διαθεσιμότητας συστοιχιών Kubernetes της υποδομής Rancher (1)

State	Name	Version	Provider	Machines	Age
Active	cluster01	v1.24.9+rke2r2	Harvester RKE2	2	125 days
Active	cluster02	v1.24.9+k3s2	Harvester K3s	3	126 days
Active	coreos	v1.24.10+k3s1	Imported K3s	2	108 days
Active	local	v1.24.10+k3s1	Local K3s	1	126 days

Στιγμιότυπο 40. Δοκιμή υψηλής διαθεσιμότητας συστοιχιών Kubernetes της υποδομής Rancher (2)

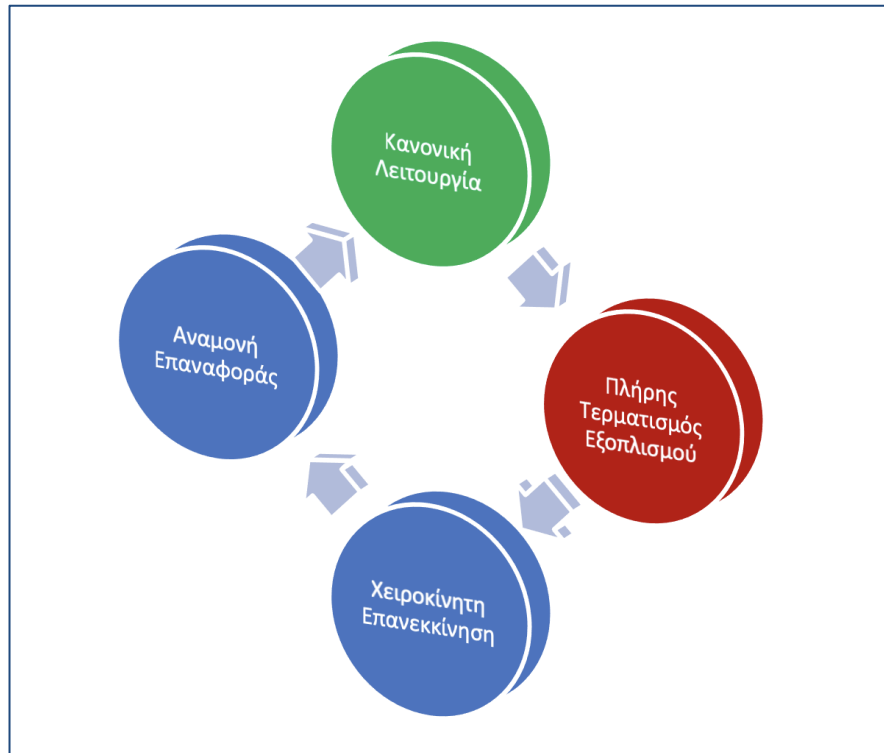
Μία ακόμα δοκιμή πραγματοποιήθηκε για την μελέτη της διαθεσιμότητας της πλατφόρμας Rancher, η οποία, στην παρούσα υλοποίηση, φιλοξενείται στην υποδομή IaaS Harvester. Για την δοκιμή της υψηλής διαθεσιμότητας του Rancher πρέπει να ελεγχθεί είτε η υψηλή διαθεσιμότητα της Kubernetes συστοιχίας στην οποία φιλοξενείται, είτε η υψηλή διαθεσιμότητα της υποδομής (φυσικοί ή εικονικοί κόμβοι) στην οποία φιλοξενείται η Kubernetes συστοιχία. Αναφορικά με τον έλεγχο της υψηλής διαθεσιμότητας της υποδομής που φιλοξενείται η συστοιχία δεν απαιτούνται επιπλέον ενέργειες, αφού οι αντίστοιχοι έλεγχοι πραγματοποιήθηκαν με την εκτέλεση του πρώτου σεναρίου. Αντίστοιχα, η υλοποίηση του δεύτερου σεναρίου υπερκαλύπτει τον έλεγχο υψηλής διαθεσιμότητας συστοιχιών Kubernetes που έχουν αναπτυχθεί με την αξιοποίηση εικονικών μηχανών της υποδομής Harvester. Τέλος, πρέπει να επισημανθεί ότι δεν αποτελεί προϋπόθεση η φιλοξενία

της πλατφόρμας Rancher στην υποδομή Harvester, αλλά επιλέχθηκε για λόγους ενοποίησης (consolidation) και οικονομίας πόρων.

Συνοψίζοντας, οι δοκιμές επικεντρώθηκαν στην αστοχία των δύο κύριων υπηρεσιών που προσφέρει η υποδομή, δηλαδή στην παροχή υπηρεσιών φιλοξενίας εικονικών μηχανών (IaaS) και περιεκτών (CaaS). Οι δοκιμές υλοποιήθηκαν με τέτοιο τρόπο ώστε να εκτιμηθεί η απόκριση των πλατφορμών ανεξάρτητα από την πηγή προέλευσης της αστοχίας (υλικό, δίκτυο, επίθεση). Αν και οι δοκιμές πραγματοποιήθηκαν σε περιορισμένο πλήθος κόμβων κατέδειξαν σαφώς τη δυνατότητα της υποδομής για παροχή υπηρεσιών υψηλής διαθεσιμότητας.

4.4.2.8 Δοκιμή ολικής απενεργοποίησης και επαναφοράς

Η διαδικασία της επαναφοράς μιας υποδομής σε κανονική λειτουργία, μετά την πλήρη απενεργοποίησή της, πρέπει να εξεταστεί, να καταγραφεί και να δοκιμαστεί πριν τεθεί σε παραγωγική λειτουργία. Αν και η πλήρης απενεργοποίηση μιας υποδομής δεν αποτελεί ούτε συχνό, ούτε επιθυμητό φαινόμενο στα σύγχρονα Κέντρα Δεδομένων, δεν μπορεί να υποτιμηθεί η κρισιμότητα επαναφοράς της υποδομής, ακόμα και αν η υποδομή αυτή απενεργοποιηθεί μόνο μία φορά, αφού η επαναφορά των υπηρεσιών και των δεδομένων μπορεί να είναι περίπλοκη ή και ακατόρθωτη.



Σχήμα 10. Σχηματική απεικόνιση διαδικασίας ολικής απενεργοποίησης και επαναφοράς της υποδομής Harvester και Rancher

Η πλήρης απενεργοποίηση της υποδομής αφορά τον τερματισμό λειτουργίας του εξοπλισμού που την φιλοξενεί. Πιο αναλυτικά, περιλαμβάνει την απενεργοποίηση των εξυπηρετητών που φιλοξενούν τη συστοιχία του Harvester καθώς και όλο τον παρελκόμενο εξοπλισμό (δικτυακούς μεταγωγείς, αποθηκευτικά συστήματα κ.α.). Η διαδικασία επαναφοράς μελετάται με την παραδοχή ότι όλος ο εξοπλισμός θα ενεργοποιηθεί ταυτόχρονα. Αυτό σημαίνει ότι πρέπει να ληφθούν υπόψιν οι χρόνοι εκκίνησης των υποδομών, η χρονική σειρά επαναφοράς και η παραμετροποίηση του χρόνου αναμονής ώστε να γίνουν διαθέσιμοι οι απαιτούμενοι πόροι. Στην προκείμενη περίπτωση οι δοκιμές εκτελέστηκαν απενεργοποιώντας τους κόμβους της συστοιχίας του Harvester και απενεργοποιώντας τις δικτυακές θύρες στις οποίες συνδέονται. Στην υλοποίηση της υποδομής δεν περιλαμβάνεται άλλος περιφερειακός εξοπλισμός – όπως αποθηκευτικά συστήματα – αφού όλοι οι διαθέσιμοι πόροι παρέχονται μέσω των εξυπηρετητών.

Τα αποτελέσματα των δοκιμών ήταν αρκετά ενθαρρυντικά, αφού σε όλες τις περιπτώσεις, η συστοιχία επανήλθε σε πλήρη λειτουργία, χωρίς την παρέμβαση των διαχειριστών.

4.4.3 OKD αξιοποιώντας την δυνατότητα του για ενσωμάτωση στο oVirt

Το OKD, παλιότερα γνωστό ως OpenShift Origin, αποτελεί την κοινοτική έκδοση του OpenShift, της ευρέως διαδεδομένης πλατφόρμας υπηρεσιών καταμεμημένων εφαρμογών και φιλοξενίας περιεκτών της Red Hat. Ως μία ανοιχτού κώδικα πλατφόρμα, το OKD επιτρέπει την ανάπτυξη, την διαχείριση και τον συνεχή έλεγχο των εφαρμογών που είναι ενσωματωμένες σε περιέκτες. Η πλατφόρμα βασίζεται στο Kubernetes, το οποίο είναι ένα σύστημα ανοιχτού κώδικα για την αυτοματοποίηση της εγκατάστασης, της κλιμάκωσης και της διαχείρισης εφαρμογών περιεκτών. Το OKD, αξιοποιώντας τις λειτουργίες του Kubernetes, επιτρέπει στους χρήστες να διαχειρίζονται με ευελιξία τους διαθέσιμους πόρους και την κατανομή του φόρτου εργασίας σε νεφούπολογιστικά περιβάλλοντα. Καθώς οι τεχνολογίες της υπολογιστικής νέφους και των περιεκτών καθορίζουν, πλέον, τον τρόπο που οι οργανισμοί αντιμετωπίζουν τον ψηφιακό μετασχηματισμό, το OKD αποτελεί ένα ιδιαίτερα χρήσιμο εργαλείο για την μετάβαση επιχειρήσεων και οργανισμών στην τεχνολογία νέφους.

Το OKD έχει τη δυνατότητα εγκατάστασης και ενσωμάτωσης του στο oVirt, επιτρέποντας την διαχείριση εικονικών μηχανών και εφαρμογών σε περιέκτες μέσα από μια ενοποιημένη πλατφόρμα. Αυτό επιτρέπει την αύξηση της αποτελεσματικότητας και της αποδοτικότητας, καθώς εξαλείφονται πολλές από τις προκλήσεις που συνδέονται με την διαχείριση ανεξάρτητων πλατφορμών φιλοξενίας υπηρεσιών νέφους. Επιπλέον, η υποστήριξη του OKD για αυτόματη κλιμάκωση εφαρμογών, συνδυαστικά με την ενσωμάτωση του στο oVirt, επιτρέπει την αναπροσαρμογή των πόρων σε πραγματικό χρόνο, δημιουργώντας ένα περιβάλλον που μπορεί να ανταποκριθεί, ευέλικτα, στις συνεχείς αλλαγές απαιτήσεις των εφαρμογών.

Η δυνατότητα αξιοποίησης της πλατφόρμας oVirt για την εγκατάσταση του OKD, σε συνδυασμό με τις διαθέσιμες υποδομές του Φορέα στην πλατφόρμα oVirt, αποτελεί έναν ιδιαίτερα ελκυστικό παράγοντα για την μελέτη της συγκεκριμένης λύσης.

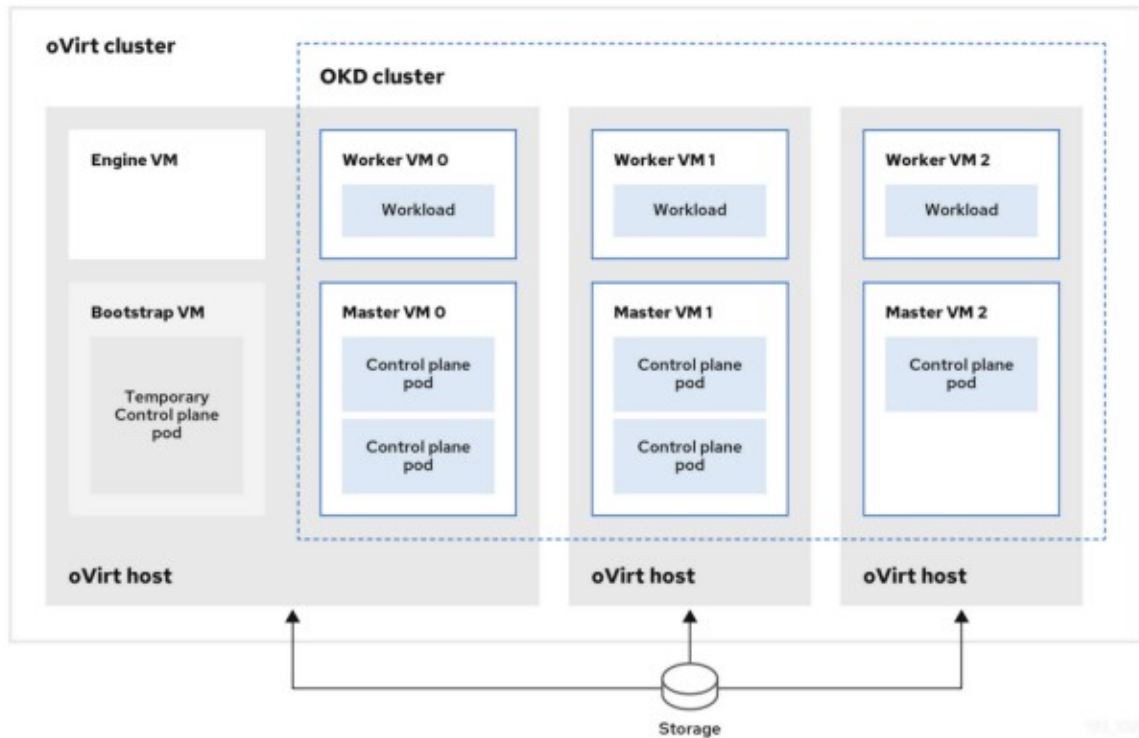
4.4.3.1 Εγκατάσταση υποδομής

Η ομάδα διαχείρισης συστημάτων και δικτύων, όντας καταρτισμένη και διαθέτοντας εμπειρία στην πλατφόρμα oVirt, προχώρησε στην εγκατάσταση της

συμβατής διανομής του OKD σε υπάρχουσα συστοιχία οVirt, που διέθετε ο Φορέας. Για την διαδικασία της εγκατάστασης μελετήθηκε η επίσημη διαθέσιμη τεκμηρίωση, η οποία μπορεί να χαρακτηριστεί ως επαρκής. Σύμφωνα με την τεκμηρίωση, επιλέχθηκε η μεθοδολογία εγκατάστασης συστοιχίας OKD (Installer-provisioned infrastructure cluster) με χαρακτηριστικά υψηλής διαθεσιμότητας, χωρίς μοναδικό σημείο αποτυχίας και με πλήρη δυνατότητα ελέγχου και αναβάθμισης από τους διαχειριστές. Πιο συγκεκριμένα, με βάση την τεκμηρίωση εγκατάστασης συστοιχίας OKD σε υποδομή οVirt, απαιτείται η εκτέλεση συγκεκριμένων ενεργειών, οι οποίες περιλαμβάνουν την προετοιμασία της εγκατάστασης, την εκτέλεση της αυτοματοποιημένης εγκατάστασης και την παραμετροποίηση της συστοιχίας.

Προετοιμασία της εγκατάστασης

Η αυτοματοποιημένη διαδικασία εγκατάστασης προϋποθέτει συγκεκριμένες δικτυακές υποδομές και εικονική ή φυσική μηχανή που θα αξιοποιηθεί αποκλειστικά για την εγκατάσταση της συστοιχίας. Οι δικτυακές υποδομές αφορούν την ανάθεση υποδικτύων που θα αξιοποιηθούν από τους κόμβους, τις εφαρμογές και το γραφικό περιβάλλον διεπαφής χρήστη, την ονοματοδοσία (DNS) της προγραμματιστικής διεπαφής (API) και των μελλοντικών εφαρμογών (*.apps) της συστοιχίας, όπως και την δυναμική IP διευθυνσιοδότηση (DHCP) των κόμβων. Παράλληλα το παραμετροποιημένο αρχείο εντολών εγκατάστασης (installer script), διατίθεται από την επίσημη διανομή του OKD, στο αποθετήριο GitHub. Η εκτέλεση του πρέπει να πραγματοποιηθεί σε υπολογιστικό μηχάνημα ανεξάρτητο της συστοιχίας, αλλά με δυνατότητες πρόσβασης στα δίκτυα και στους κόμβους της συστοιχίας που θα δημιουργηθούν, ενώ απαιτεί και την δημιουργία κατάλληλων αρχείων παραμέτρων (configuration files) αναφορικά με την προσβασιμότητα στη συστοιχία οVirt και τις επιθυμητές ρυθμίσεις (π.χ. πλήθος master & worker κόμβων) της προς εγκατάσταση συστοιχίας του OKD. Επιπρόσθετα των ρυθμίσεων αυτών, στα πλαίσια εγκατάστασης στο οVirt, είναι απαραίτητη και η μεταφόρτωση των πρότυπων εικόνων (templates) εικονικών μηχανών, που θα χρησιμοποιήσει το αρχείο εντολών εγκατάστασης για την δημιουργία των κόμβων της συστοιχίας. Τέλος, μέρος της προετοιμασίας αποτέλεσε και η διαστασιοποίηση (sizing) των απαραίτητων πόρων, σε επεξεργαστική ισχύ, σε μνήμη και σε αποθηκευτικό χώρο, κάτι καθόλου αμελητέο σε περιορισμένες, σε πόρους, ιδιωτικές υποδομές, αφού η ελάχιστη υποδομή απαιτεί τουλάχιστον 7 εικονικές μηχανές.



Εικόνα 4. Αρχιτεκτονική συστοιχίας OKD σε συστοιχία oVirt

Εγκατάσταση της συστοιχίας

Η ομάδα, αφού επέλεξε την δημιουργία μίας εικονικής μηχανής ως το υπολογιστικό σύστημα μέσα από το οποίο θα εκτελεστεί το αρχείο εγκατάστασης, ενσωμάτωσε τα απαραίτητα αρχεία παραμέτρων (ovirt-config.yaml, install-config.yaml) και προχώρησε στην πραγματοποίηση της εγκατάστασης. Η αυτοματοποιημένη διαδικασία της εγκατάστασης περιλαμβάνει μία σειρά από εργασίες που πρέπει να ολοκληρωθούν ώστε να δημιουργηθεί επιτυχώς η OKD συστοιχία. Οι εργασίες αυτές εμπεριέχουν την δημιουργία εικονικής μηχανής εκκίνησης (bootstrap), την δημιουργία των master κόμβων, την εκκίνηση λειτουργίας της προγραμματιστικής διεπαφής (API), την εκκίνηση λειτουργίας του ελεγκτή εισόδου (ingress controller) των εφαρμογών, την δημιουργία των worker κόμβων, την μεταφορά του ελεγκτή εισόδου στους worker κόμβους και τέλος την αυτοκαταστροφή της εικονικής μηχανής εκκίνησης. Αν και η διαδικασία είναι αυτοματοποιημένη, απαιτήθηκαν αρκετές διορθώσεις και τροποποιήσεις μέχρι να ολοκληρωθεί επιτυχώς και να αντιμετωπιστούν οι αστοχίες που παρουσιάστηκαν.

Compute > Virtual Machines

Vms: okd

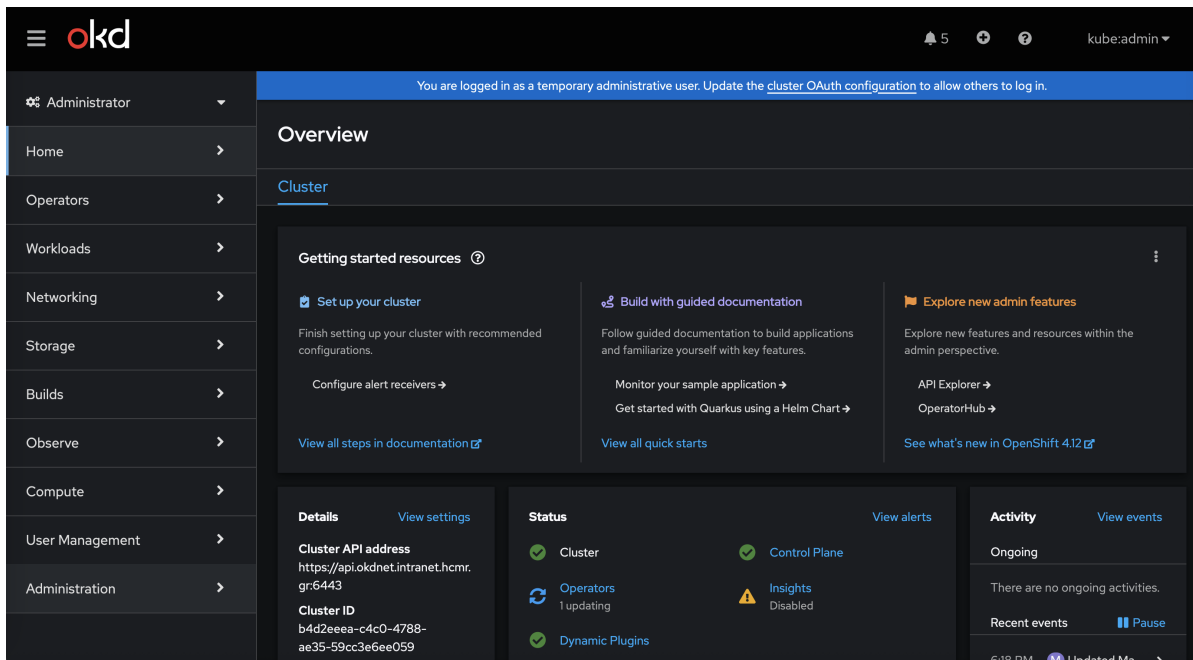
Name	Comment	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics	Status	Description
okdnet-97ftn-master-0		10.5.1.125 10.0.40...	vmcloud73	AnavissosDC	39%	21%	0%	None	Up	
okdnet-97ftn-master-1		10.5.1.104 169.25...	vmcloud72	AnavissosDC	46%	29%	0%	None	Up	
okdnet-97ftn-master-2		10.5.1.117 169.25...	vmcloud74	AnavissosDC	34%	20%	0%	None	Up	
okdnet-97ftn-worker-87d8q		10.5.1.105 169.25...	vmcloud72	AnavissosDC	34%	18%	0%	VNC	Up	
okdnet-97ftn-worker-jx9bp		10.5.1.134 169.25...	vmcloud71	AnavissosDC	24%	16%	0%	VNC	Up	
okdnet-97ftn-worker-kpjtj		10.5.1.171 169.25...	vmcloud73	AnavissosDC	15%	5%	0%	VNC	Up	
okdprovcloud		10.5.1.109 fe80::5...	vmcloud71	AnavissosDC	14%	0%	0%	VNC	Up	okd provisioning VM fr...

Στιγμιότυπο 41. Εικονικές μηχανές συστοιχίας OKD στο oVirt

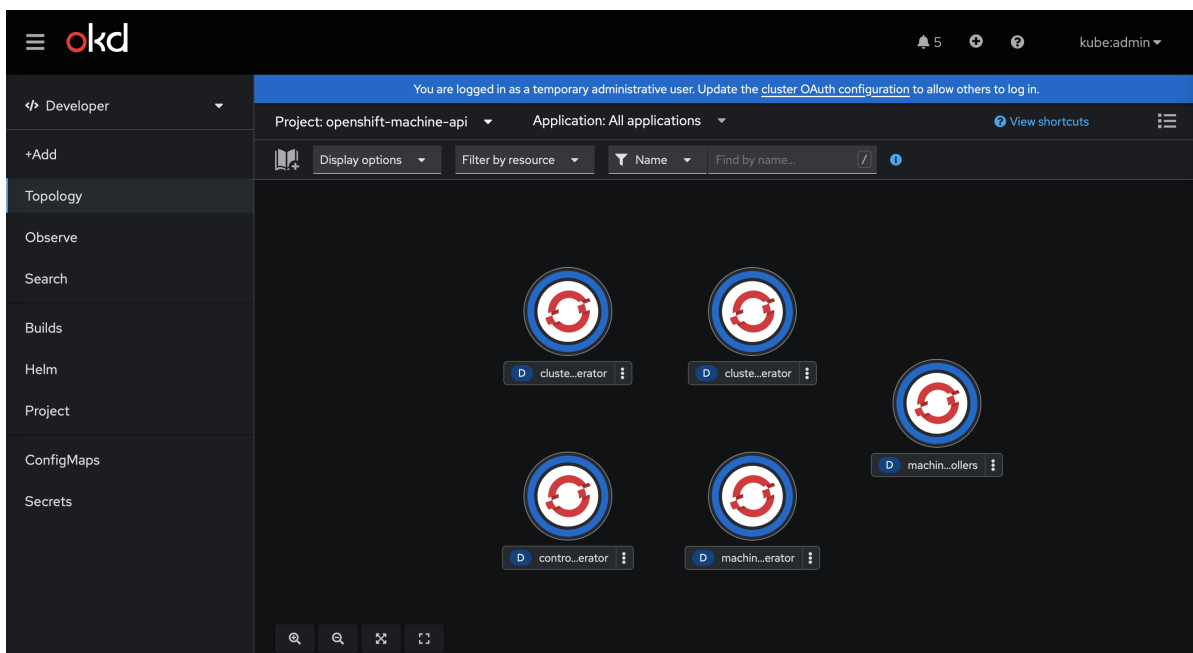
Μετά την ολοκλήρωση της εκτέλεσης του αρχείου εντολών γραμμής (script) εγκατάστασης και αφού ελέγχθηκαν τα αρχεία καταγραφής της εγκατάστασης (logs) για τυχόν σφάλματα ή ελλείψεις, πραγματοποιηθήκαν δοκιμές βασικής λειτουργικότητας της συστοιχίας. Οι δοκιμές αυτές περιλάμβαναν την επιτυχή σύνδεση με το API του OKD (curl), την πρόσβαση στη γραφική διεπαφή χρήστη και τον έλεγχο ορθής λειτουργίας των εικονικών μηχανών στις οποίες φιλοξενούνταν η συστοιχία.

4.4.3.2 Προσβασιμότητα στο UI

Μετά την ολοκλήρωση των βασικών ελέγχων αναφορικά με την επιτυχή εγκατάσταση της συστοιχίας OKD, πραγματοποιήθηκε είσοδος στο γραφικό περιβάλλον, σύμφωνα με τις οδηγίες που εμπεριείχε το αρχείο καταγραφής της εγκατάστασης. Πιο αναλυτικά, το αρχείο καταγραφής περιέγραφε τη διαδικασία πρόσβασης στο γραφικό περιβάλλον με την ενσωμάτωση σε αυτό του URL, του ονόματος χρήστη και του κωδικού πρόσβασης.



Στιγμιότυπο 42. Γραφικό περιβάλλον OKD - κονσόλα διαχειριστή



Στιγμιότυπο 43. Γραφικό περιβάλλον OKD - κονσόλα προγραμματιστή

Εν συνεχεία, η ομάδα περιηγήθηκε στις διάφορες διεπαφές που διαθέτει το γραφικό περιβάλλον, το οποίο διαχωρίζεται σε δύο κύριες κατηγορίες διεπαφών, του Διαχειριστή (Administrator Console) της συστοιχίας και του Προγραμματιστή (Developer Console) (βλ. Στιγμιότυπο 44-43).

Η διεπαφή του Administrator του OKD, γνωστή και ως Admin Console, είναι το σημείο αναφοράς για τη διαχείριση των υποδομών, των κόμβων, των διαφόρων

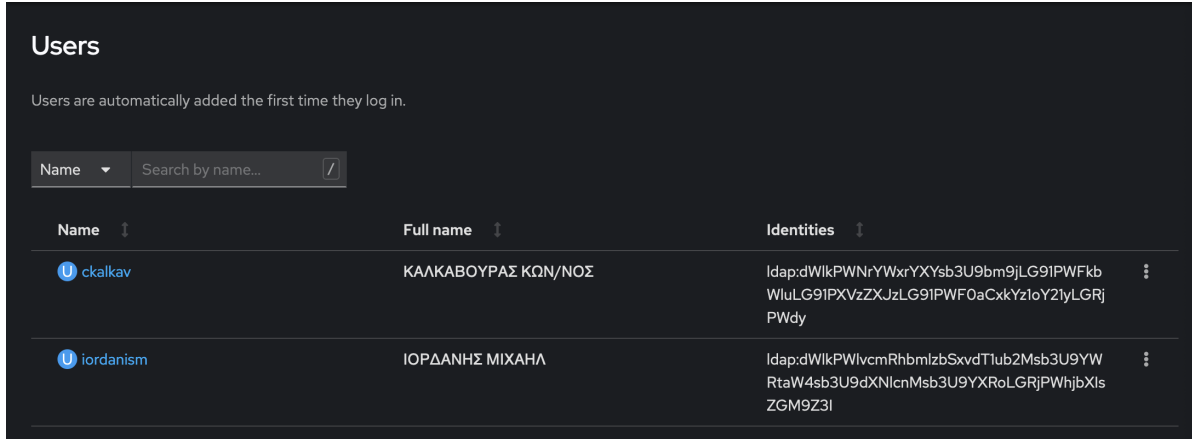
επιπέδων ασφαλείας, της δικτύωσης και των υπόλοιπων πόρων της OKD συστοιχίας. Στην αρχική διεπαφή προβάλλεται μια επισκόπηση της κατάστασης της συστοιχίας, που περιλαμβάνει πληθώρα πληροφοριών όπως τον αριθμό των κόμβων, την κατάσταση των pods, την χρήση των πόρων και τα τελευταία συμβάντα (events). Επιπλέον, περιλαμβάνει διεπαφές για τη διαχείριση διαφόρων στοιχείων του συστήματος όπως των αποθετηρίων εφαρμογών (Operators - Registries), των εργασιών (Workloads), του δικτύου, των συστημάτων αποθήκευσης, των κόμβων, της παρακολούθησης (Observer - Monitoring), των χρηστών κ.α.. Συνολικά, το OKD Administrator Console αποτελεί ένα ολοκληρωμένο εργαλείο για τη διαχείριση της υποδομής και τη λειτουργία μιας OKD συστοιχίας.

Αντίστοιχα η διεπαφή του προγραμματιστή περιέχει όλες τις λειτουργικότητες που απαιτούνται για την διαχείριση του κύκλου ζωής μιας εφαρμογής και την εφαρμογή της σύγχρονης πρακτικής συνεχούς ανάπτυξης και παράδοσης (CI/CD). Πιο αναλυτικά, η διεπαφή του Developer γνωστή και ως Developer Console, προσφέρει έναν γραφικό εργαλείο για τους προγραμματιστές ώστε να διαχειρίζονται τις εφαρμογές, τα αντίγραφα τους, τις υπηρεσίες, τα αποθετήρια, τα pipelines και τα builds. Επιπλέον προσφέρεται επισκόπηση των έργων (projects), ενώ διατίθεται και μια διεπαφή προβολής της τοπολογίας των εφαρμογών (topology), που εμφανίζει ένα γραφικό διάγραμμα των εφαρμογών και των αλληλεπιδράσεών τους. Επίσης, υπάρχει η διεπαφή της υπηρεσίας Καταλόγου (Catalog), μέσα από την οποία ο προγραμματιστής μπορεί να αναζητήσει και να εγκαταστήσει λογισμικά και εφαρμογές ή επιπλέον αποθετήρια και εργαλεία. Το Developer Console δεν στερείται της δυνατότητας παρακολούθησης πόρων, απόδοσης και προβολής αρχείων καταγραφής, κάτι ιδιαίτερα κρίσιμο κατά την ανάπτυξη εφαρμογών και αποτελεί ένα ισχυρό και ευέλικτο εργαλείο για την διαχείριση και την παρακολούθηση των εφαρμογών από τους προγραμματιστές.

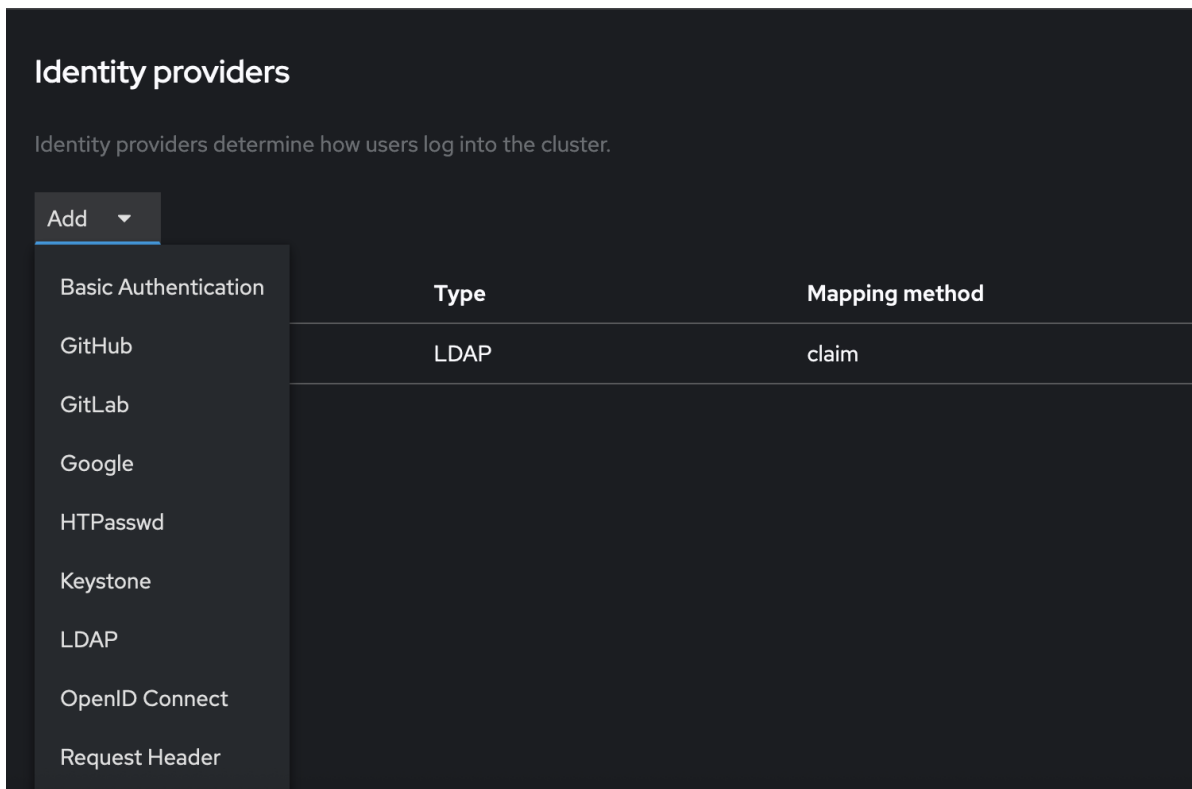
4.4.3.3 Δημιουργία χρηστών και διαχείριση πόρων

Το OKD επιτρέπει την διαχείριση των χρηστών, τόσο μέσω της διεπαφή γραμμής εντολών (oc - cli), όσο και μέσω της γραφικής διεπαφής του Διαχειριστή (Admin Console). Για την αυθεντικοποίηση των χρηστών παρέχεται ανεξάρτητο υποσύστημα το οποίο υποστηρίζει πληθώρα παρόχων ταυτότητας (identity

providers) προσφέροντας και τη δυνατότητα αυθεντικοποίησης μέσω της μεθόδου SSO. Η ομάδα διαχείρισης συστημάτων και δικτύων συνέδεσε τη συστοιχία OKD με την υπηρεσία καταλόγου και αυθεντικοποίησης OpenLDAP που διαθέτετε ήδη ο Φορέας και αξιοποιώντας την δημιούργησε χρήστες για τις δοκιμές της.



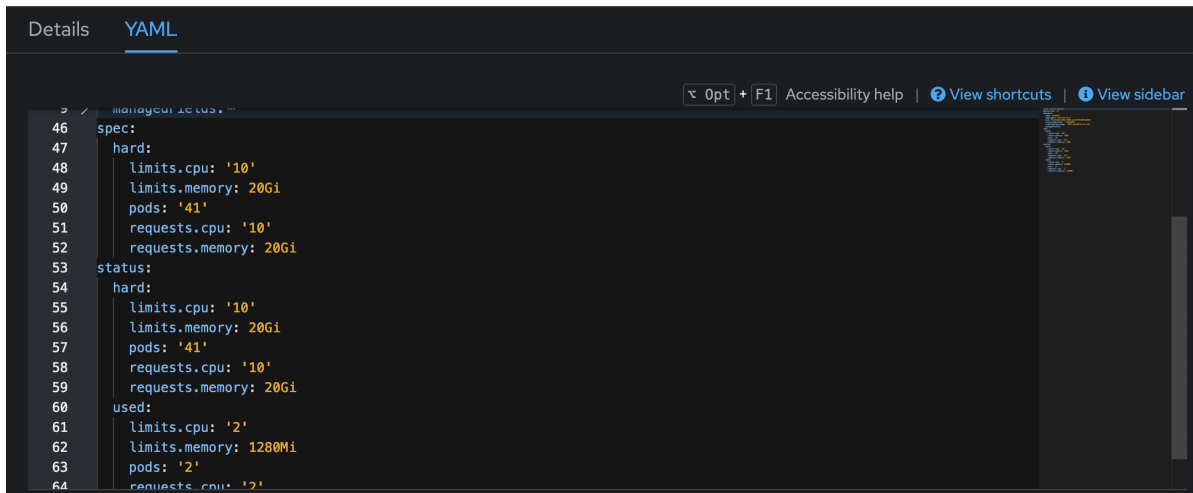
Στιγμιότυπο 45. OKD: Διεπαφή προβολής χρηστών



Στιγμιότυπο 46. OKD: Διεπαφή προσθήκης μηχανισμού αυθεντικοποίησης

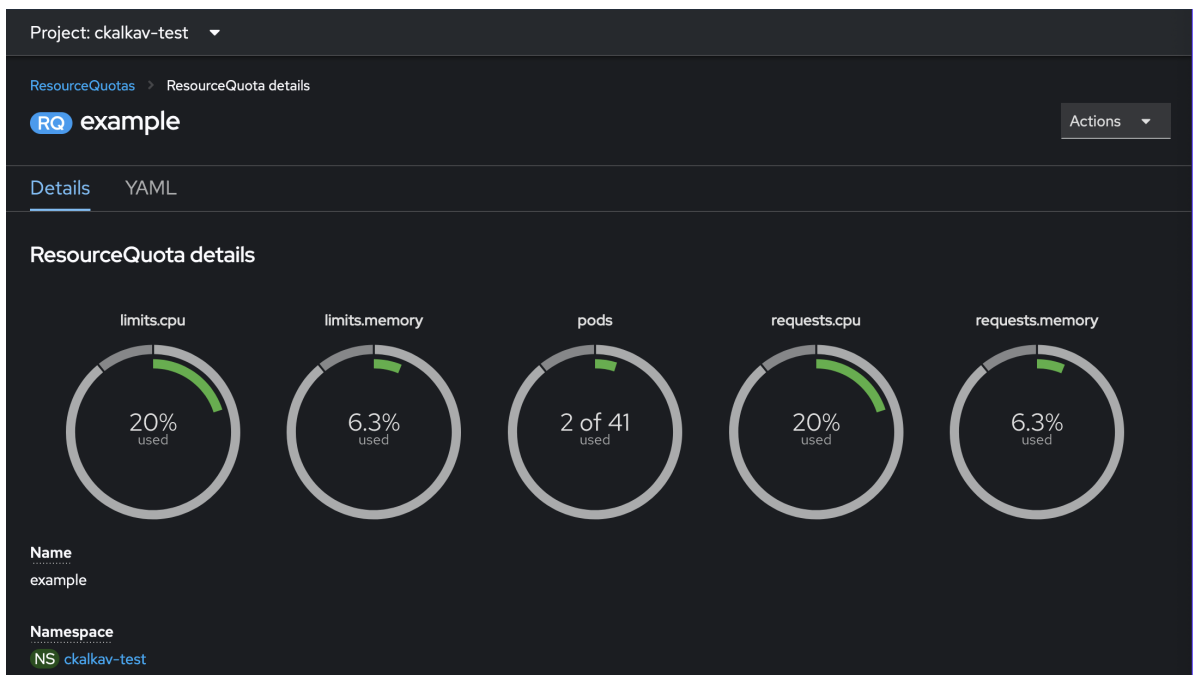
Η ανάθεση πόρων σε χρήστες στο OKD επιτυγχάνεται μέσω της χρήσης των Resource Quotas και των Limit Ranges. Τα Quotas επιτρέπουν στον διαχειριστή να ορίζει μέγιστα όρια για την κατανάλωση πόρων σε ένα namespace (Project), ενώ τα Limit Ranges επιτρέπουν τον ορισμό των ελάχιστων και μέγιστων πόρων που

μπορούν να χρησιμοποιηθούν από pods, περιέκτες ή μονάδες αποθήκευσης (volumes). Η ομάδα αξιοποίησε τη δυνατότητα ανάθεσης πόρων (Resource Quotas) ανά έργο (Project / Namespace) ώστε να ελέγξει την ορθή λειτουργία τους.



```
46 spec:
47   hard:
48     limits.cpu: '10'
49     limits.memory: 20Gi
50     pods: '41'
51     requests.cpu: '10'
52     requests.memory: 20Gi
53 status:
54   hard:
55     limits.cpu: '10'
56     limits.memory: 20Gi
57     pods: '41'
58     requests.cpu: '10'
59     requests.memory: 20Gi
60 used:
61   limits.cpu: '2'
62   limits.memory: 1280Mi
63   pods: '2'
64   requests.cpu: '2'
```

Στιγμιότυπο 47. OKD: Ανάθεση πόρων σε έργο



Στιγμιότυπο 48. OKD: Χρήση πόρων

4.4.3.4 Εγκατάσταση δοκιμαστικής εφαρμογής

Αντίστοιχα με τις δοκιμές που πραγματοποιήθηκαν στην υποδομή Rancher-Harvester, η ομάδα διαχείρισης συστημάτων και δικτύων δοκίμασε δύο διαφορετικούς τρόπους εγκατάστασης εφαρμογών. Ο ένας τρόπος πραγματεύεται

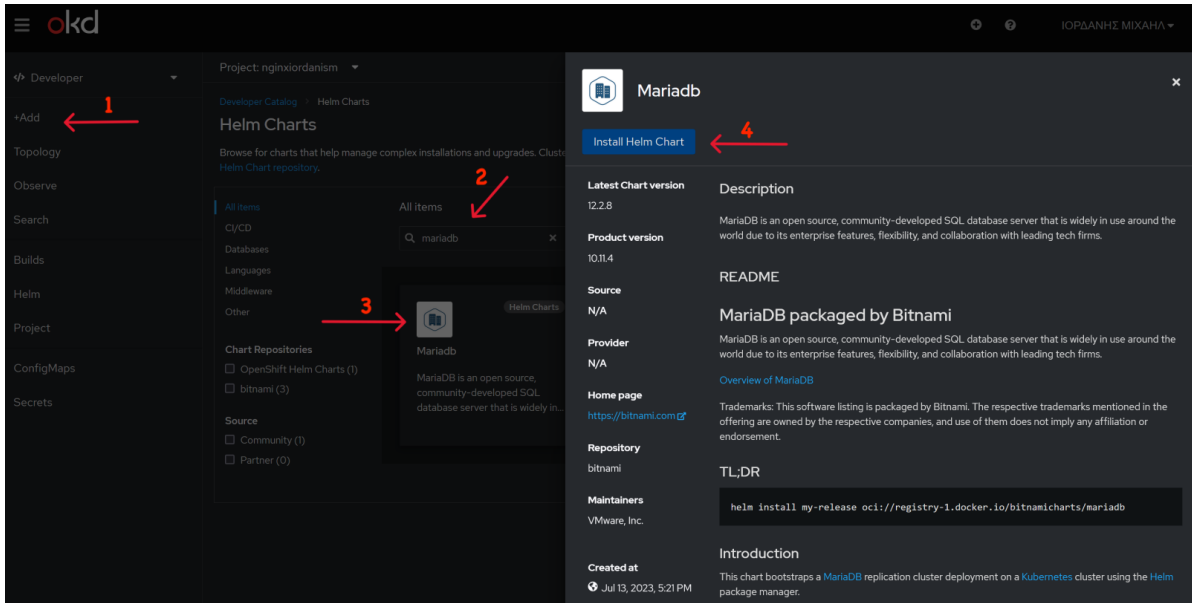
την εγκατάσταση εφαρμογής με την αξιοποίηση αποθετηρίων, ενώ ο δεύτερος τρόπος αφορά την εγκατάσταση εφαρμογής μέσω της εγκατάστασης εικόνων περιεκτών από ιδιωτικά ή δημόσια αποθετήρια και δημιουργία των απαραίτητων Kubernetes “υπηρεσιών” (Kubernetes Services).

Η πρώτη μέθοδος καταδεικνύει πώς το OKD προσφέρει μια πιο αυτοματοποιημένη προσέγγιση στην εγκατάσταση εφαρμογών, που απλοποιεί τη διαδικασία για τον χρήστη, καθώς απαιτεί λιγότερη εξοικείωση και παρέμβαση. Αντίθετα, η δεύτερη μέθοδος απαιτεί από τον χρήστη περισσότερες γνώσεις και εκτενέστερη τροποποίηση παραμέτρων, που την καθιστά λιγότερο φιλική προς χρήστες που είναι λιγότερο εξοικειωμένοι με την διαχείριση περιεκτών.

Οι δοκιμές υλοποιήθηκαν μέσω χρήστη ο οποίος αυθεντικοποιείται από την υπηρεσία καταλόγου OpenLDAP και στον οποίο έχουν ανατεθεί, από τον διαχειριστή του συστήματος, συγκεκριμένοι υπολογιστικοί πόροι.

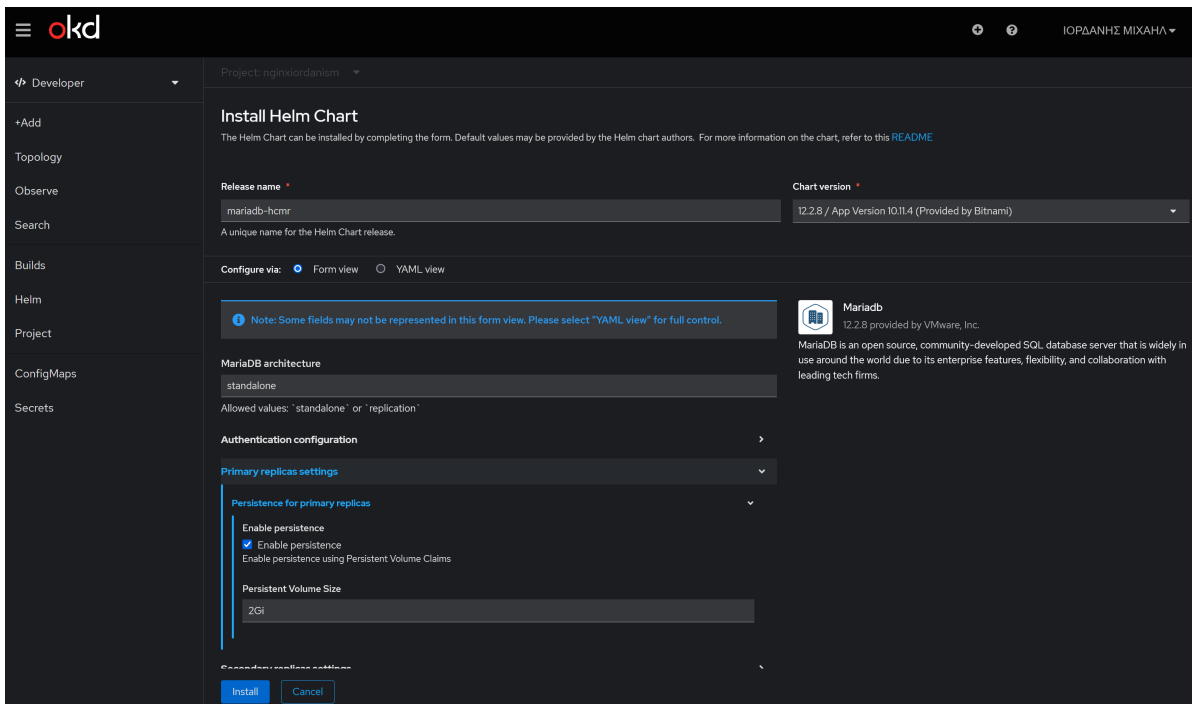
Εγκατάσταση βάσης δεδομένων MariaDB μέσω αποθετηρίου “Helm”

Η διαδικασία της εγκατάστασης της βάσης δεδομένων υλοποιήθηκε με χρήση των διαθέσιμων διεπαφών που διαθέτει η διεπαφή Προγραμματιστή (Developer Console), και αξιοποιώντας τα προεγκατεστημένα αποθετήρια εφαρμογών. Για την απομόνωση και διαχωρισμό των χρησιμοποιούμενων πόρων, δημιουργήθηκε ξεχωριστό project (namespace) εντός του οποίου ομαδοποιούνται όλοι οι πόροι που απαιτούνται από την εφαρμογή. Τα βήματα που απαιτούνται για την εγκατάσταση της δοκιμαστικής εφαρμογής MariaDB προκύπτουν εύκολα και με σαφήνεια από το γραφικό περιβάλλον διευκολύνοντας και καθοδηγώντας τον χρήστη (βλ. Στιγμιότυπο 49).

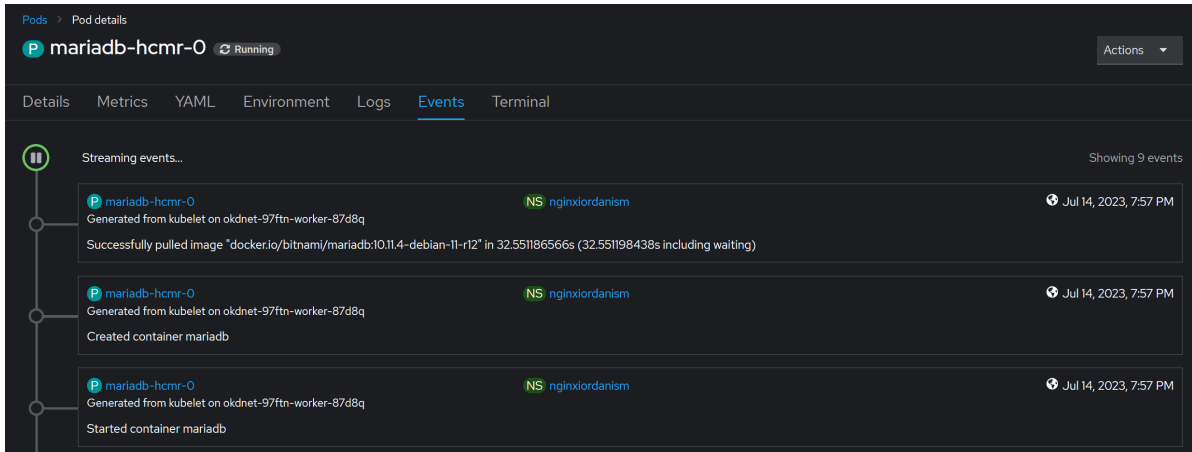


Στιγμιότυπο 50. Εγκατάσταση βάσης δεδομένων MariaDB μέσω αποθετηρίου Helm (1)

Αντίστοιχα, η διεπαφή με τη μορφή φόρμας για την συμπλήρωση των απαιτούμενων πληροφοριών της εφαρμογής, παρέχει μια φιλικότερη μορφή σε σχέση με την επεξεργασία του αντίστοιχου yaml αρχείου, ενώ διασφαλίζει την ορθή συμπλήρωση, ακόμα και για πιο άπειρους χρήστες, αφού παρουσιάζονται ευκρινώς τα απαραίτητα πεδία για την επιτυχή δημιουργία της εφαρμογής.

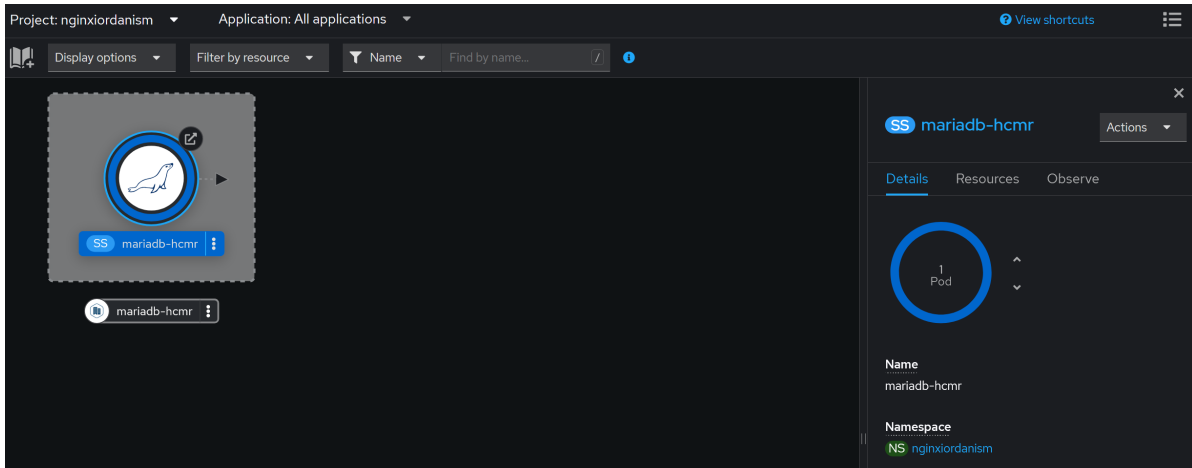


Στιγμιότυπο 51. Εγκατάσταση βάσης δεδομένων MariaDB μέσω αποθετηρίου Helm (2)

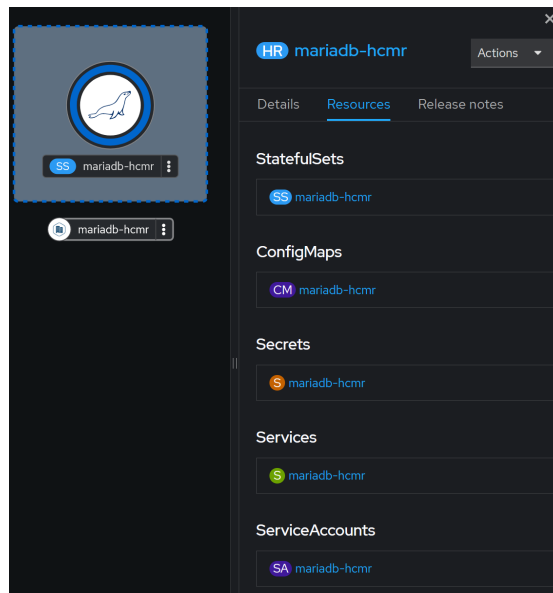


Στιγμιότυπο 52. Παρακολούθηση γεγονότων εγκατάστασης βάσης δεδομένων MariaDB

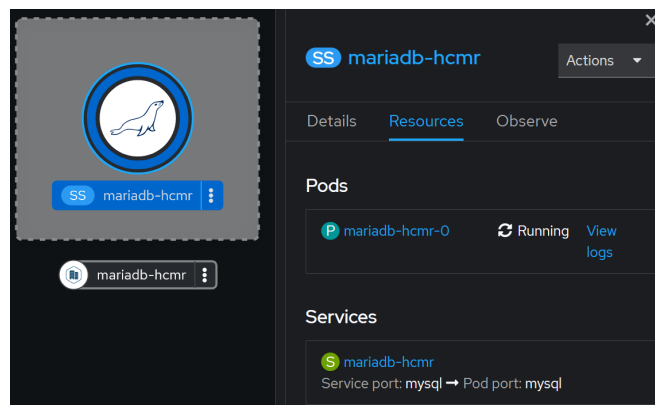
Η εφαρμογή αφού εγκατασταθεί, προβάλλεται στην διεπαφή τοπολογίας του Προγραμματιστή είτε με την μορφή εικονιδίων, είτε με την μορφή λίστας και περιλαμβάνει λειτουργικότητες πλοήγησης στους διάφορους πόρους (pods, services, secrets κ.α.) και επιπλέον πληροφορίες (events, metrics κ.α.) της εφαρμογής.



Στιγμιότυπο 53. Γραφική απεικόνιση βάσης δεδομένων MariaDB στην OKD διεπαφή



Στιγμιότυπο 54. Προβολή των πόρων της εφαρμογής MariaDB στη διεπαφή του OKD



Στιγμιότυπο 55. Έλεγχος εγκατάστασης βάσης δεδομένων MariaDB

Η ομάδα διαχείρισης συστημάτων και δικτύων συνδέθηκε στη βάση δεδομένων και εκτέλεσε βασικές εντολές για τον έλεγχο της ορθής λειτουργίας της υπηρεσίας.

```

$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 174
Server version: 10.11.4-MariaDB Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

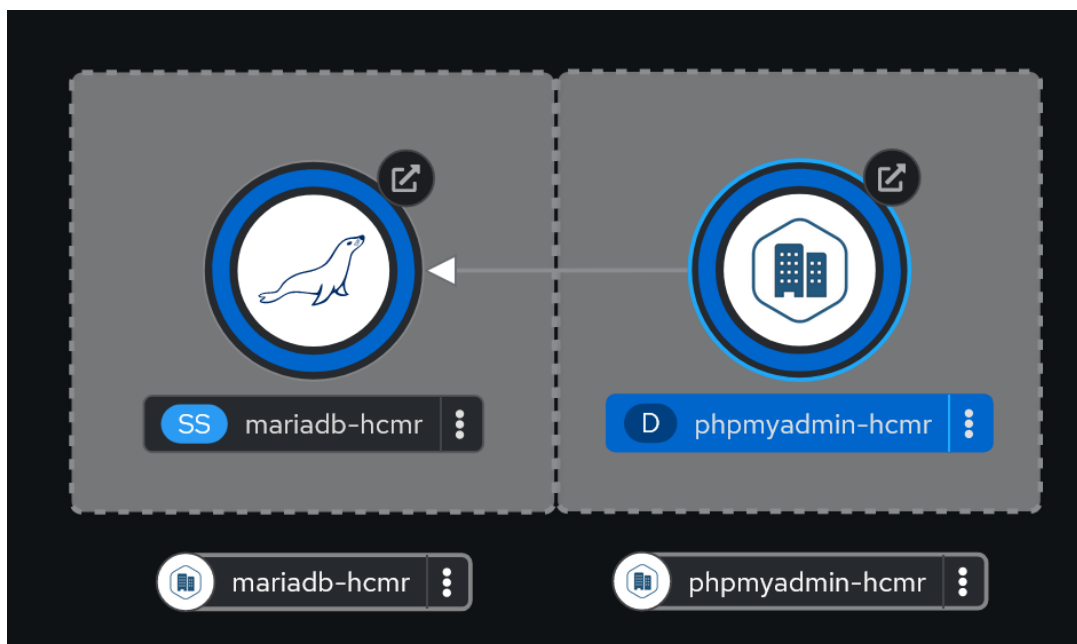
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| my_database |
| mysql |
| performance_schema |
| sys |
| test |
+-----+
6 rows in set (0.001 sec)

MariaDB [(none)]> use test;
Database changed
MariaDB [test]> show tables;
Empty set (0.000 sec)

MariaDB [test]> quit
Bye
$
    
```

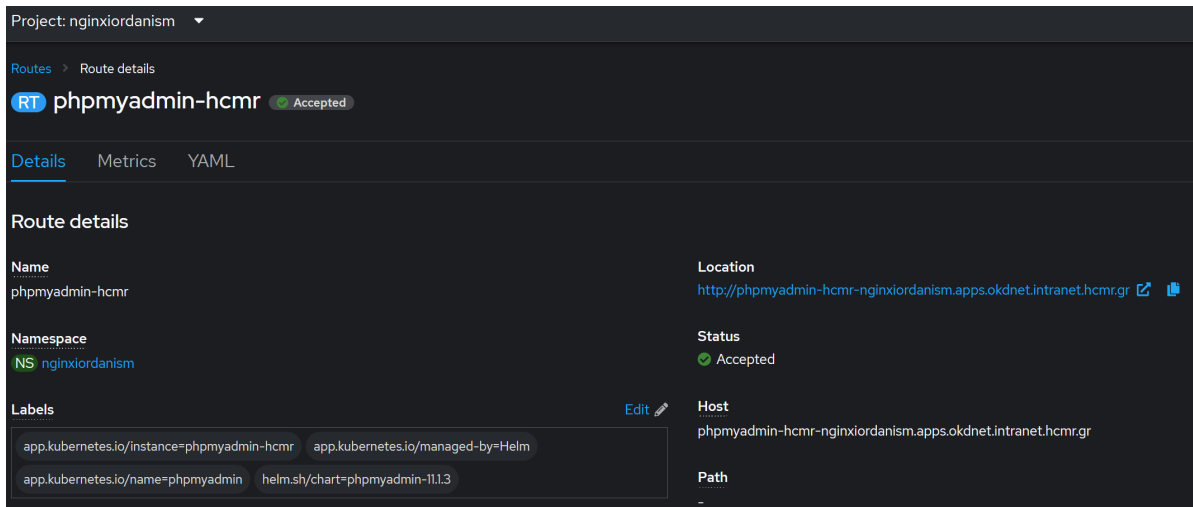
Στιγμιότυπο 56. Έλεγχος ορθής λειτουργίας βάσης δεδομένων MariaDB

Σε αντιστοιχία με την υποδομή Rancher, η αυτοματοποιημένη εγκατάσταση εφαρμογών δεν παρέχει δυνατότητα πρόσβασης στην υπηρεσία από εξωτερικά δίκτυα. Στην δοκιμή αυτή επιλέχθηκε η εγκατάσταση της βοηθητικής εφαρμογής phpMyAdmin ώστε να διασφαλιστεί η προσβασιμότητα στην βάση δεδομένων μέσω διαδικτυακής εφαρμογής (web).

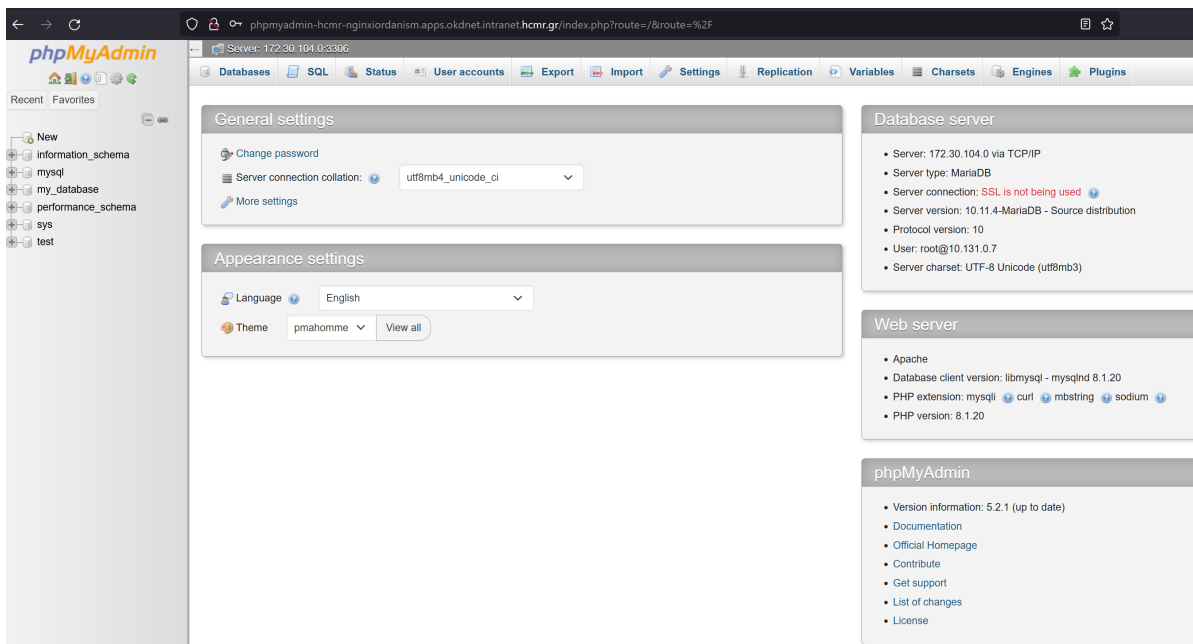


Στιγμιότυπο 57. Εγκατάσταση της βοηθητικής εφαρμογής phpMyAdmin (1)

Η εγκατάσταση της εφαρμογής phpMyAdmin υλοποιήθηκε αντίστοιχα, αξιοποιώντας τα διαθέσιμα αποθετήρια του OKD, παράλληλα με την προσθήκη του κατάλληλου route για την επικοινωνία της με εξωτερικά δίκτυα.



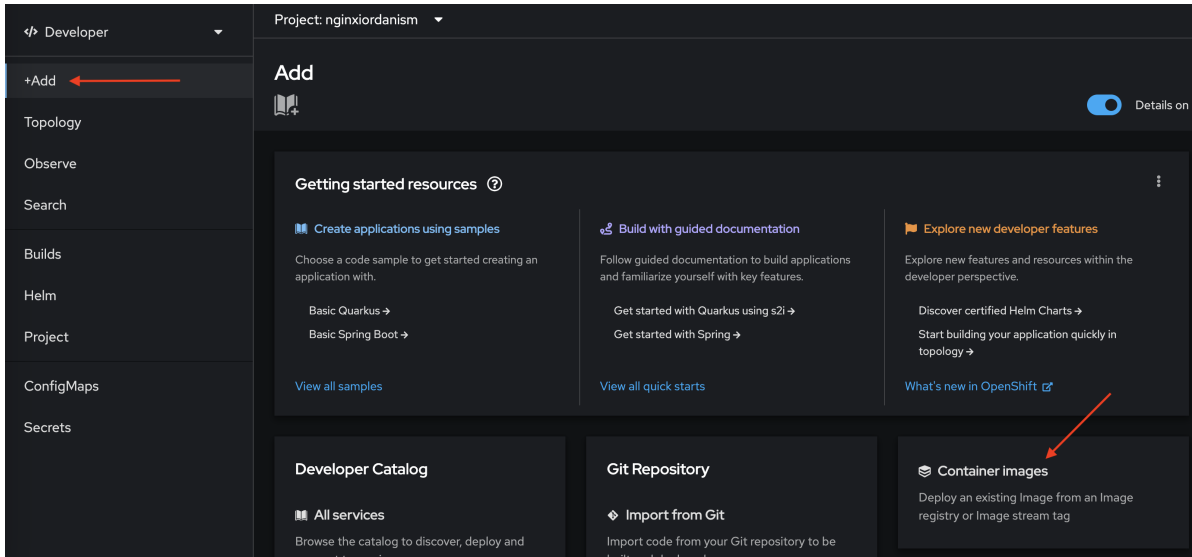
Στιγμιότυπο 58. Εγκατάσταση της βοηθητικής εφαρμογής phpMyAdmin (2)



Στιγμιότυπο 59. Έλεγχος ορθής λειτουργίας phpMyAdmin

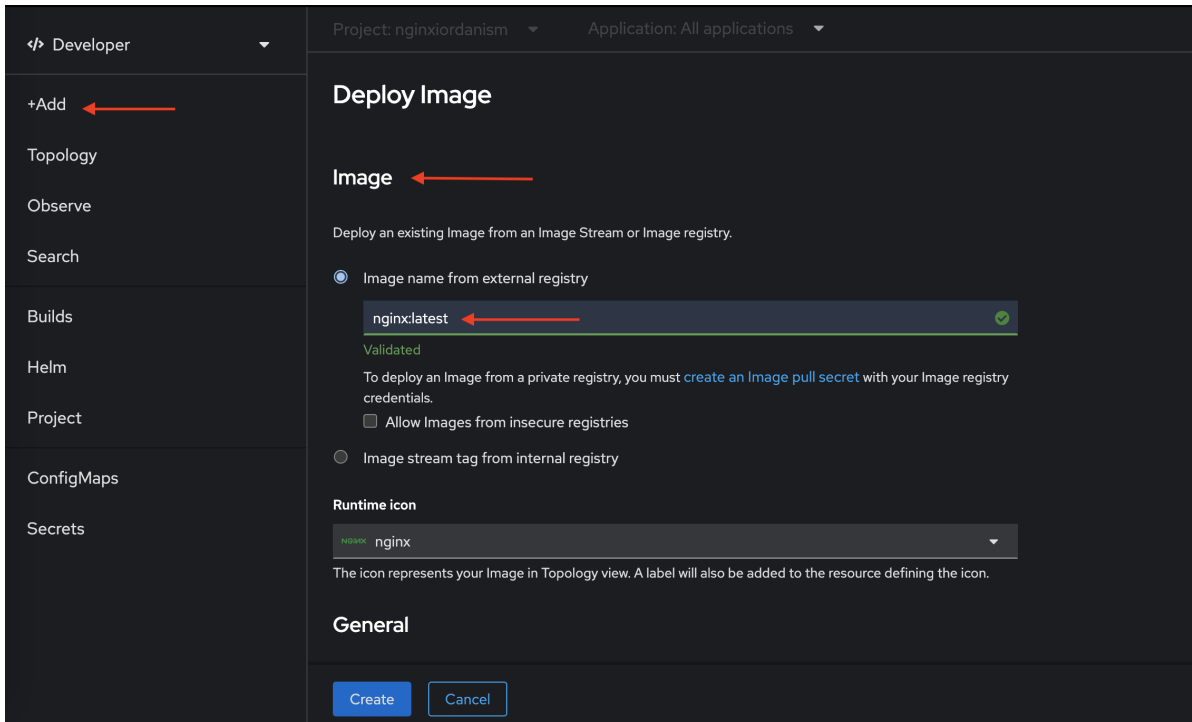
Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου

Η διεπαφή Προγραμματιστή (Developer Console) υποστηρίζει την δημιουργία εφαρμογών μέσω χρήσης περιεκτών από εξωτερικά ή εσωτερικά αποθετήρια εικόνων περιέκτη, όπως επίσης και την προσθήκη περιεκτών σε υπάρχουσες εφαρμογές με αντίστοιχη μέθοδο.



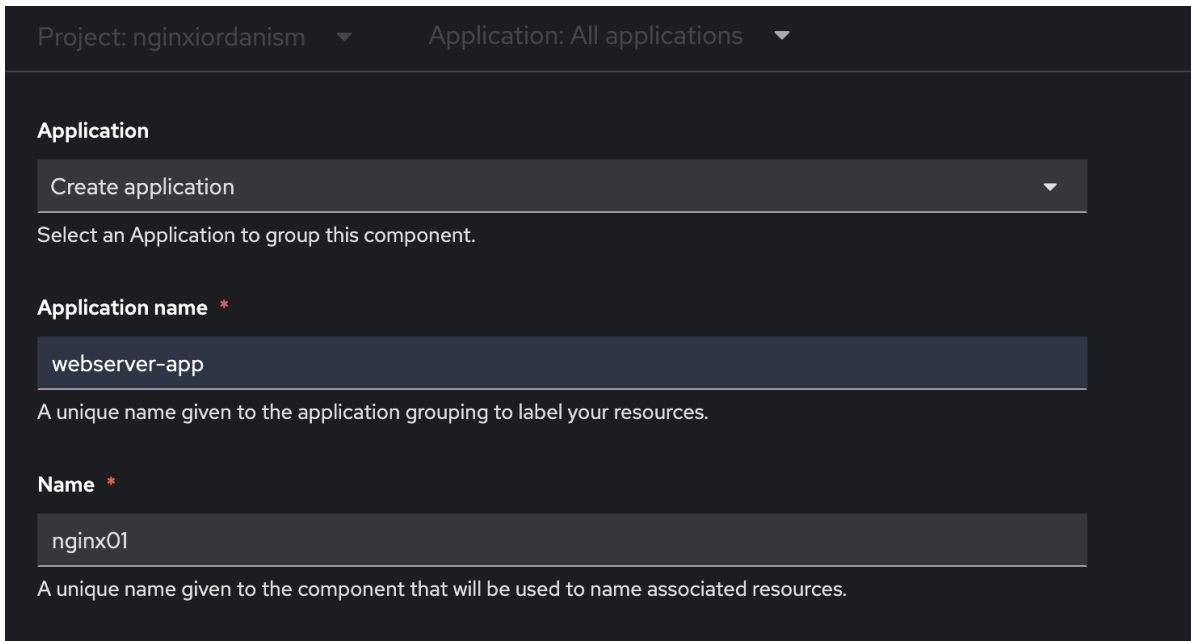
Στιγμιότυπο 60. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (1)

Η ομάδα διαχείρισης συστημάτων και δικτύων, αξιοποιώντας τη λειτουργία αυτή και το δημόσιο αποθετήριο Docker Hub, δημιούργησε μία νέα εφαρμογή, ενσωματώνοντας της τον περιέκτη εξυπηρετητή δικτύου nginx.



Στιγμιότυπο 61. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (2)

Με την συμπλήρωση των διαθέσιμων πεδίων της φόρμας, που διαθέτει η διεπαφή, μεταξύ άλλων παραμετροποιήθηκαν χαρακτηριστικά της εφαρμογής όπως η θύρα επικοινωνίας και το hostname της εφαρμογής.



The screenshot shows a configuration interface with a dark theme. At the top, there are two dropdown menus: 'Project: nginxjordanism' and 'Application: All applications'. Below these, the 'Application' section contains a dropdown menu with 'Create application' selected and the instruction 'Select an Application to group this component.' The 'Application name' section has a text input field containing 'webserver-app' and the instruction 'A unique name given to the application grouping to label your resources.' The 'Name' section has a text input field containing 'nginx01' and the instruction 'A unique name given to the component that will be used to name associated resources.'

Στιγμιότυπο 62. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (3)

Project: nginxordanism Application: All applications

Advanced options

Target port

80

Target port for traffic.

Create a route
Exposes your component at a public URL

Hide advanced Routing options

Hostname

webserver01

Public hostname for the route. If not specified, a hostname is generated.

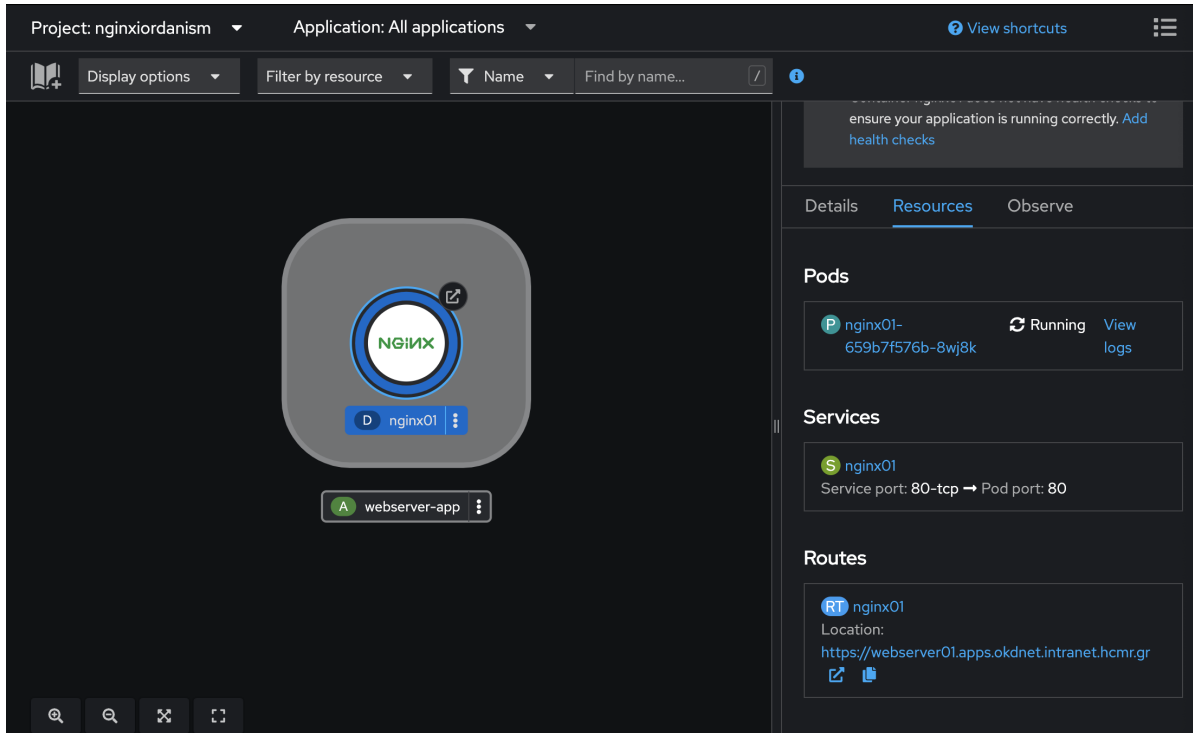
Path

/

Path that the router watches to route traffic to the service.

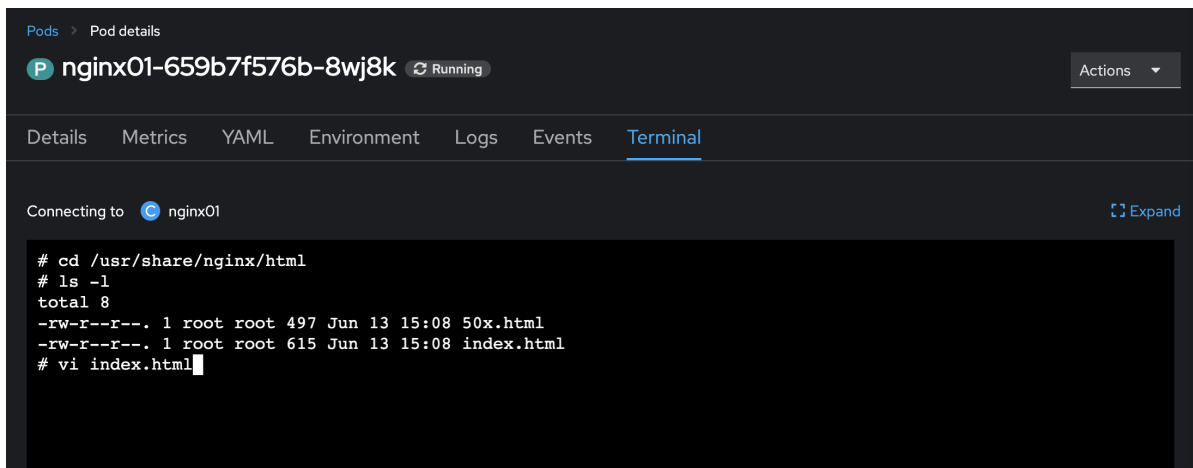
Στιγμιότυπο 63. Εγκατάσταση εξυπηρετητή δικτύου (webserver) nginx μέσω δημόσιου αποθετηρίου (4)

Με την ολοκλήρωση της εγκατάστασης, δεσμεύτηκαν και δημιουργήθηκαν οι απαιτούμενοι πόροι, όπως το Pod, το Service και το Route.

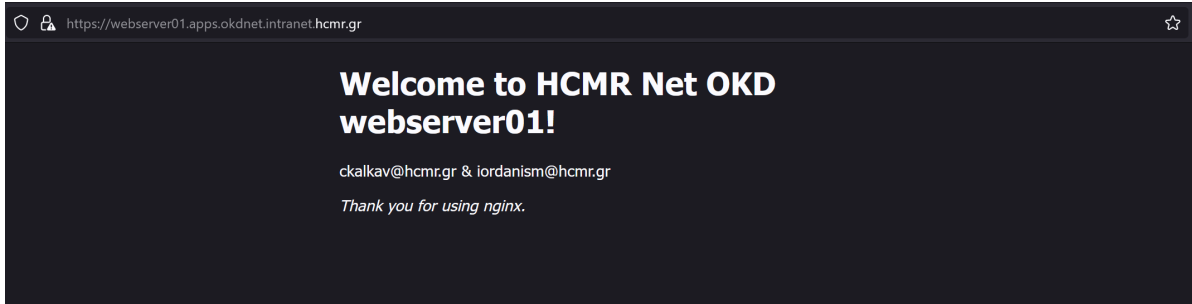


Στιγμιότυπο 64. Προβολή των πόρων της εφαρμογής εξυπηρετητή δικτύου (webserver) nginx

Στην συνέχεια, στα πλαίσια της δοκιμής, πραγματοποιήθηκε τροποποίηση της αρχικής σελίδας του εξυπηρετητή δικτύου, μέσω της χρήσης κονσόλας που προσφέρεται από την διεπαφή του OKD, ώστε να ελεγχθεί περαιτέρω η προσβασιμότητα στα αρχεία παραμετροποίησης και η λειτουργικότητα της εφαρμογής.



Στιγμιότυπο 65. Έλεγχος ορθής λειτουργίας της εφαρμογής εξυπηρετητή δικτύου (webserver) nginx (1)



Στιγμιότυπο 66. Έλεγχος ορθής λειτουργίας της εφαρμογής εξυπηρετητή δικτύου (webserver) nginx (2)

Εγκατάσταση WordPress εφαρμογής μέσω αρχείου yaml

Η ομάδα διαχείρισης συστημάτων και δικτύων, αναγνωρίζοντας την δυναμική της OKD πλατφόρμας συνδυαστικά με την υπάρχουσα υποδομή oVirt, προέβη και σε τρίτη δοκιμή που αφορούσε την εγκατάσταση της ευρέως διαδεδομένης εφαρμογής WordPress με τη χρήση αρχείων yaml και αξιοποιώντας την επίσημη τεκμηρίωση του Kubernetes [70]. Η υλοποίηση της εφαρμογής, αυτής, πραγματοποιήθηκε με χρήση της γραμμής εντολών, του εργαλείου γραμμής εντολών “oc” και του Kubernetes εργαλείου διαμόρφωσης kustomize, χωρίς τη συμμετοχή του γραφικού περιβάλλοντος του OKD.

Για την εγκατάσταση της εφαρμογής, δημιουργήθηκαν και παραμετροποιήθηκαν κατάλληλα τρία αρχεία yaml. Τα αρχεία αυτά περιγράφουν το deployment της εφαρμογής του WordPress, το deployment της βάσης δεδομένων που απαιτείται και άλλες απαιτούμενες παραμέτρους.

```
[root@okdnet-97ftn-master-0 wordpress-kust1]# ll
-rw-r--r--. 1 root root 137 May  8 19:48 kustomization.yaml
-rw-r--r--. 1 root root 1193 May  8 19:52 mysql-deployment.yaml
-rw-r--r--. 1 root root 1329 May 10 17:16 wordpress-deployment.yaml
[root@okdnet-97ftn-master-0 wordpress-kust1]#
```

Τα βήματα της εγκατάστασης περιλάμβαναν:

- Την είσοδο στην OKD πλατφόρμα μέσω της γραμμής εντολών
- Την επιλογή του κατάλληλου project (namespace)

```
kubectl config set-context --current --namespace=nginxiordanism
```

```
#or
```

```
oc project nginxiordanism
```

- Την διαμόρφωση των τριών yaml αρχείων

```
[root@okdnet-97ftn-master-0 core]# cat wordpress-kust1/kustomization.yaml
```

```
secretGenerator:
```

```
- name: mysql01-pass
```

```
  literals:
```

```
  - password=test1234
```

```
resources:
```

```
- mysql-deployment.yaml
```

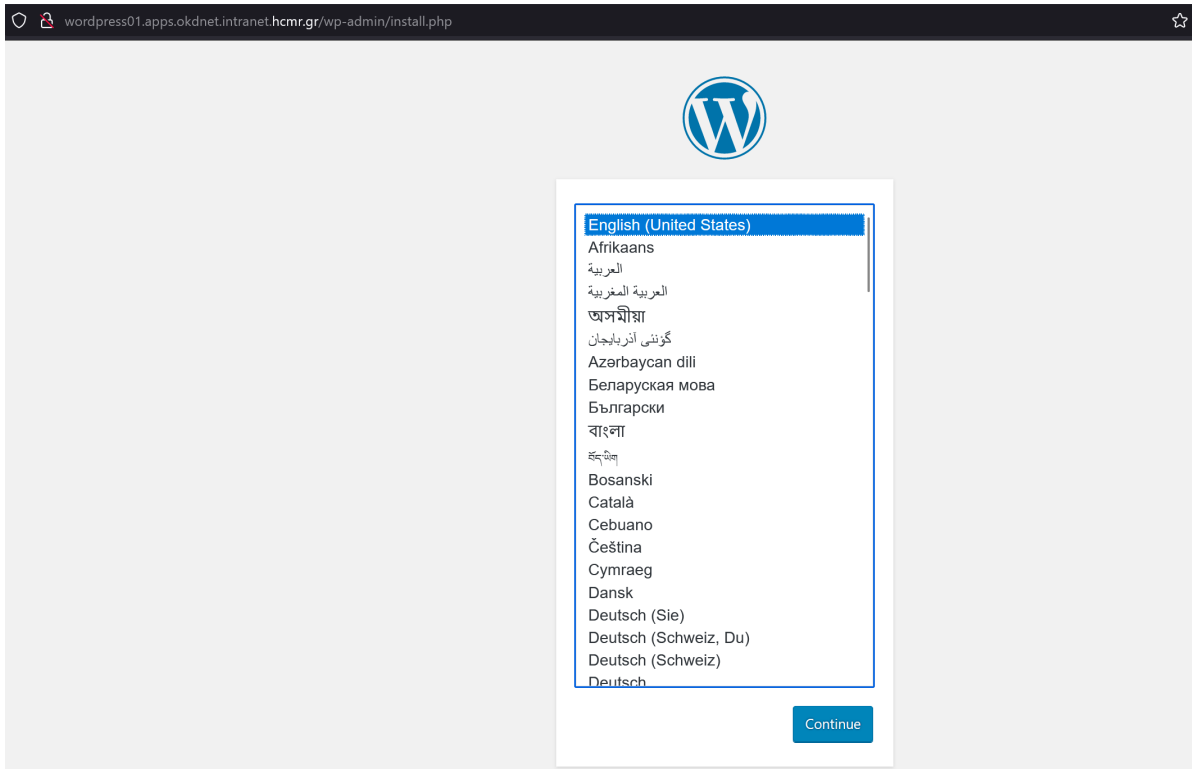
```
- wordpress-deployment.yaml
```

```
[root@okdnet-97ftn-master-0 core]#
```

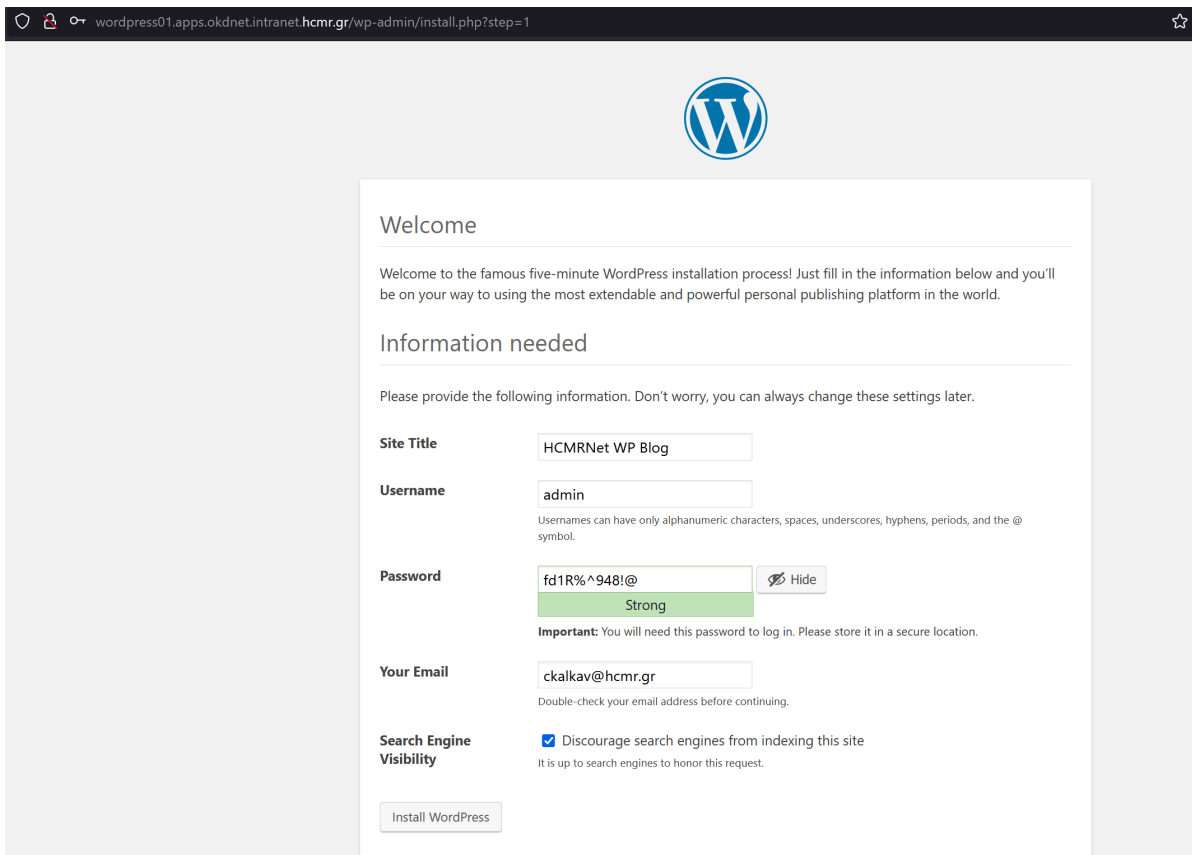
- Την εκτέλεση της εντολής `oc apply` με την παράμετρο “-k” που επεξεργάζεται κατάλληλα διαμορφωμένο (kustomize) κατάλογο αρχείων

```
[root@okdnet-97ftn-master-0 core]# oc apply -k ./wordpress-kust1/
```

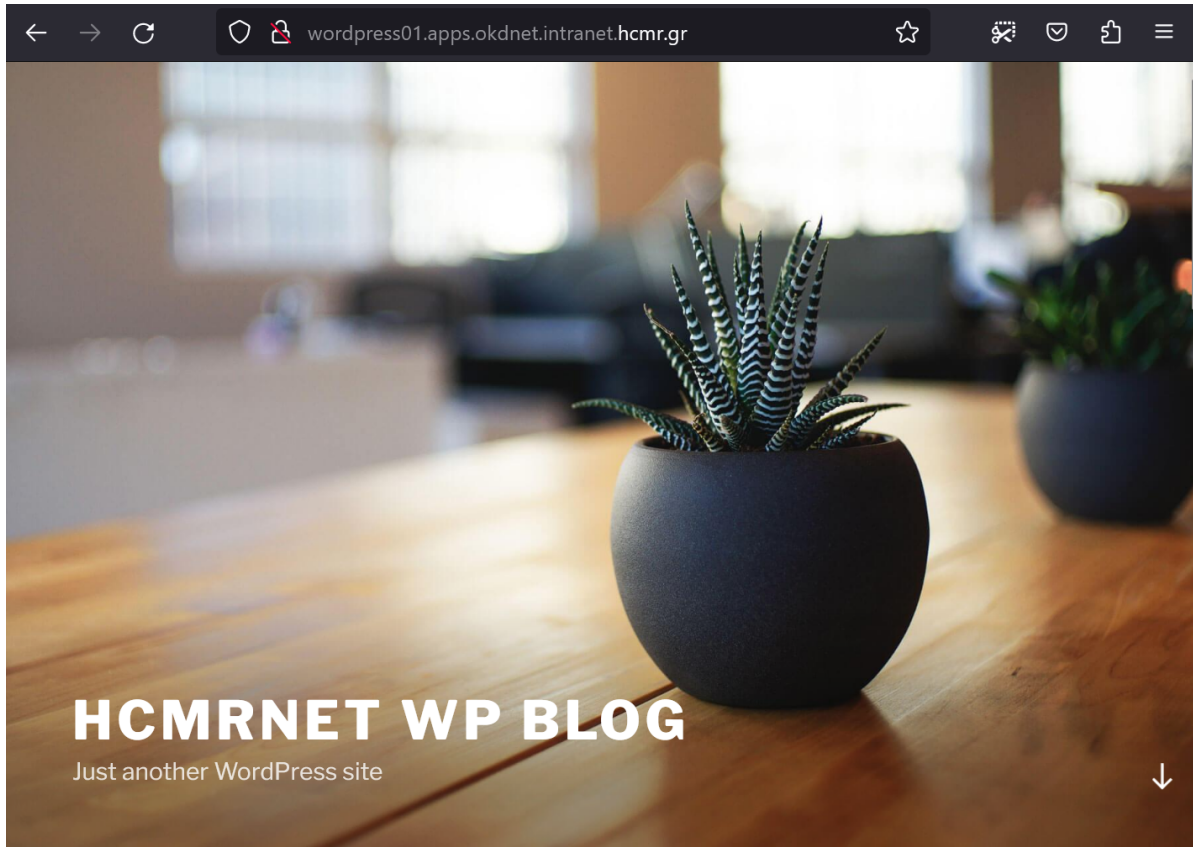
Μετά την ολοκλήρωση της εγκατάστασης και την δημιουργία υπηρεσίας δρομολόγησης (route), η πλατφόρμα WordPress είναι διαθέσιμη μέσω φυλλομετρητή (browser), για την περαιτέρω παραμετροποίηση της.



Στιγμιότυπο 67. Εγκατάσταση WordPress εφαρμογής μέσω αρχείου yaml (1)



Στιγμιότυπο 68. Εγκατάσταση WordPress εφαρμογής μέσω αρχείου yaml (2)



Στιγμιότυπο 69. Εγκατάσταση WordPress εφαρμογής μέσω αρχείου yaml (3)

4.4.3.5 Επέκταση υποδομής

Η υποδομή OKD μπορεί να επεκταθεί μέσω διάφορων τεχνικών. Οι δύο κύριες μεθοδολογίες επέκτασης, που εφαρμόζονται στους περισσότερους τύπους συστοιχιών, περιλαμβάνουν την οριζόντια (scale out) και την κάθετη (scale up) επέκταση.

Η οριζόντια επέκταση στο OKD σημαίνει την προσθήκη περισσότερων κόμβων στη συστοιχία. Αυτό επιτρέπει στη συστοιχία να χειριστεί περισσότερα pods και, κατά συνέπεια, περισσότερες εφαρμογές. Η οριζόντια επέκταση είναι ιδιαίτερα χρήσιμη για τη διαχείριση αυξημένης κίνησης ή την αντιμετώπιση αύξησης του φόρτου εργασίας, δεδομένου ότι υποστηρίζεται επαρκώς η αυτοματοποίηση της. Στο OKD, μπορεί να αξιοποιηθεί η οντότητα των Machine Sets ώστε να αυτοματοποιηθεί η διαδικασία οριζόντιας επέκτασης. Τα Machine Sets επιτρέπουν την δημιουργία νέων κόμβων με ειδικά προκαθορισμένες ρυθμίσεις, διευκολύνοντας την οριζόντια επέκταση.

Η κάθετη επέκταση στο OKD επιτυγχάνεται με την προσθήκη περισσότερων πόρων σε έναν υπάρχων κόμβο. Αυτό μπορεί να περιλαμβάνει την προσθήκη περισσότερης μνήμης, επεξεργαστικής ισχύος ή αποθηκευτικού χώρου. Η κάθετη επέκταση μπορεί να βελτιώσει την απόδοση των εφαρμογών που εκτελούνται σε ένα κόμβο, αλλά υπάρχει καθορισμένο όριο στις δυνατότητες επέκτασης κάθε μεμονωμένου κόμβου. Το όριο επέκτασης κάθε κόμβου εξαρτάται από τους διαθέσιμους πόρους του υπολογιστικού συστήματος στο οποίο εκτελείται, είτε πρόκειται για εικονική μηχανή, είτε για φυσική μηχανή.

Και οι δύο μέθοδοι επέκτασης έχουν τα δικά τους πλεονεκτήματα και μπορούν να χρησιμοποιηθούν ανάλογα με τις ανάγκες των εφαρμογών ή του φόρτου εργασίας, που καλούνται να εξυπηρετήσουν. Πρέπει να επισημανθεί πως σε ιδιωτικές περιορισμένες υποδομές, κρίσιμο παράγοντα αποτελούν οι διαθέσιμοι πόροι και οι δυνατότητες επέκτασης της συστοιχίας (IaaS) στην οποία εγκαθίσταται το OKD. Κατά αντιστοιχία με τη συστοιχία OKD, η συστοιχία του oVirt μπορεί να επεκταθεί, και αυτή, με τις δύο προαναφερθείσες μεθολογίες, δηλαδή την προσθήκη επιπλέον κόμβων ή την αύξηση των πόρων στους υπάρχοντες κόμβους. Σε κάθε περίπτωση, η διαλειτουργικότητα που επιτυγχάνεται μεταξύ OKD και oVirt, διευκολύνει την συνολική διαχείριση των συστοιχιών, επιτρέπει την αυτόματη κλιμάκωση και προσφέρει, με βελτιστοποιημένο τρόπο, μια ενοποιημένη διεπαφή για την παρακολούθηση και την διαθεσιμότητα των πόρων.

Η ομάδα διαχείρισης συστημάτων και δικτύων για τον έλεγχο των δυνατοτήτων επέκτασης της OKD συστοιχίας και της ορθής λειτουργίας τους πραγματοποίησε δοκιμές τόσο οριζόντιας όσο και κάθετης επέκτασης. Οι δοκιμές οριζόντιας επέκτασης περιλάμβαναν την επέκταση υπάρχοντος συνόλου από worker κόμβους (MachineSet), με την προσθήκη επιπρόσθετων κόμβων, όπως και την δημιουργία νέου συνόλου από worker κόμβους με διαφορετικά χαρακτηριστικά σε πόρους.

	Name	Comment	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics	Status	Description
▲	okdnet-97ftn-master-0		10.5.1.125 10.0.40...	vmcloud73	AnavissosDC	42%	37%	0%	None	Up	
▲	okdnet-97ftn-master-1		10.5.1.104 169.25...	vmcloud72	AnavissosDC	51%	32%	0%	None	Up	
▲	okdnet-97ftn-master-2		10.5.1.117 169.25...	vmcloud74	AnavissosDC	39%	28%	0%	None	Up	
▲	okdnet-97ftn-worker-87d8q		10.5.1.105 169.25...	vmcloud72	AnavissosDC	31%	22%	0%	VNC	Up	
▲	okdnet-97ftn-worker-jx9bp		10.5.1.134 169.25...	vmcloud71	AnavissosDC	23%	21%	0%	VNC	Up	
▲	okdnet-97ftn-worker-kpjtl		10.5.1.171 169.25...	vmcloud73	AnavissosDC	15%	5%	0%	VNC	Up	
▲	okdnet-97ftn-worker-mz8pm				AnavissosDC	--	--	--	None	Image Locked	
▲	okdprovcloud		10.5.1.109 fe80::5...	vmcloud71	AnavissosDC	15%	0%	0%	VNC	Up	okd provisioning VM fr...

Στιγμιότυπο 70. Οριζόντια επέκταση υποδομής (1)

Project: openshift-machine-api

MachineSet details

MS okdnet-97ftn-worker

Details YAML Machines Events

Create machine

Name Search by name... /

Name	Node	Phase	Provider state	Region	Availability zone
okdnet-97ftn-worker-87d8q	okdnet-97ftn-worker-87d8q	Provisioned as node	up	-	-
okdnet-97ftn-worker-jx9bp	okdnet-97ftn-worker-jx9bp	Provisioned as node	up	-	-
okdnet-97ftn-worker-kpjtl	okdnet-97ftn-worker-kpjtl	Provisioned as node	up	-	-
okdnet-97ftn-worker-mz8pm		Provisioning	-	-	-

Στιγμιότυπο 71. Οριζόντια επέκταση υποδομής (2)

Compute > Virtual Machines

Vms: okd

New Edit Run Suspend Shutdown Reboot Console Create Snapshot Migrate

Name	Comment	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics	Status
okdnet-97ftn-master-0		10.5.1.125 10.0.40.87 169.254.169.2 fe80::b49e:f2e5:249:4b2e	vmcloud73	AnavissosDC	45%	31%	0%	None	Up
okdnet-97ftn-master-1		10.5.1.104 169.254.169.2 fe80::2963:7715:7ca3:fa05	vmcloud72	AnavissosDC	54%	38%	0%	None	Up
okdnet-97ftn-master-2		10.5.1.117 169.254.169.2 fe80::c96d:cb4c:bd2c:424a	vmcloud74	AnavissosDC	40%	34%	0%	None	Up
okdnet-97ftn-worker-87d8q		10.5.1.105 169.254.169.2 10.0.40.88 fe80::aacd:3de1:fa38:28ce	vmcloud72	AnavissosDC	31%	23%	0%	VNC	Up
okdnet-97ftn-worker-jx9bp		10.5.1.134 169.254.169.2 fe80::7f8:c8fcc:13:fd87	vmcloud71	AnavissosDC	25%	11%	0%	VNC	Up
okdnet-97ftn-worker-kpjtl		10.5.1.171 169.254.169.2 fe80::dce8:7eb5:3b9e:1f88	vmcloud73	AnavissosDC	16%	6%	0%	VNC	Up
okdnet-97ftn-worker-mz8pm		10.5.1.156 169.254.169.2 fe80::7f3a:168a:74ab:c883	vmcloud73	AnavissosDC	8%	36%	0%	VNC	Up
okdprovccloud		10.5.1.109 fe80::546f83fffeb3:a	vmcloud71	AnavissosDC	15%	0%	0%	VNC	Up

Στιγμιότυπο 72. Οριζόντια επέκταση υποδομής (3)

Nodes

Filter Name Search by name... /

Name	Status	Roles	Pods	Memory	CPU	Filesystem	Created	Instance type
okdnet-97ftn-master-0	Ready	control-plane, master	32	6.61 GiB / 15.57 GiB	0.818 cores / 4 cores	49.82 GiB / 119.8 GiB	Apr 1, 2023, 10:38 PM	-
okdnet-97ftn-master-1	Ready	control-plane, master	67	8.39 GiB / 15.57 GiB	0.957 cores / 4 cores	28.77 GiB / 119.8 GiB	Apr 1, 2023, 10:36 PM	-
okdnet-97ftn-master-2	Ready	control-plane, master	44	5.93 GiB / 15.57 GiB	0.622 cores / 4 cores	21.99 GiB / 119.8 GiB	Apr 1, 2023, 10:38 PM	-
okdnet-97ftn-worker-87d8q	Ready	worker	27	4.56 GiB / 15.57 GiB	0.573 cores / 4 cores	38.07 GiB / 119.8 GiB	Apr 1, 2023, 11:43 PM	-
okdnet-97ftn-worker-jx9bp	Ready	worker	19	3.5 GiB / 15.57 GiB	0.472 cores / 4 cores	27.73 GiB / 119.8 GiB	Apr 1, 2023, 11:44 PM	-
okdnet-97ftn-worker-kpjtl	Ready	worker	22	1.72 GiB / 15.57 GiB	0.186 cores / 4 cores	17.23 GiB / 119.8 GiB	Apr 1, 2023, 11:47 PM	-
okdnet-97ftn-worker-mz8pm	Ready	worker	15	1.24 GiB / 15.57 GiB	0.314 cores / 4 cores	8.61 GiB / 119.8 GiB	Jun 24, 2023, 1:08 PM	-

Στιγμιότυπο 73. Κάθετη επέκταση υποδομής (1)

Οι δοκιμές κάθετης επέκτασης της υποδομής OKD πραγματοποιήθηκαν με την επέκταση σε πόρους, επιλεγμένων workers (nodes). Η διαδικασία περιλάμβανε τη

μετατροπή της κατάστασης του κόμβου σε κατάσταση συντήρησης (maintenance), την απενεργοποίηση του, την μεταβολή των πόρων από το διαχειριστικό περιβάλλον του οVirt, την επανεκκίνηση του κόμβου και την επαναφορά του σε κατάσταση κανονικής λειτουργίας.

Name	Status	Roles	Pods	Memory	CPU	Filesystem	Created	Instance type
okdnet-97ftn-master-0	Ready	control-plane, master	45	5.79 GiB / 15.57 GiB	0.751 cores / 4 cores	57.2 GiB / 119.8 GiB	Apr 1, 2023, 10:38 PM	-
okdnet-97ftn-master-1	Ready	control-plane, master	67	6.48 GiB / 15.57 GiB	0.770 cores / 4 cores	25.39 GiB / 119.8 GiB	Apr 1, 2023, 10:36 PM	-
okdnet-97ftn-master-2	Ready	control-plane, master	31	5.44 GiB / 15.57 GiB	0.772 cores / 4 cores	20.92 GiB / 119.8 GiB	Apr 1, 2023, 10:38 PM	-
okdnet-97ftn-worker-87dbq	Ready	worker	28	4.11 GiB / 15.57 GiB	0.416 cores / 4 cores	30.03 GiB / 119.8 GiB	Apr 1, 2023, 11:43 PM	-
okdnet-97ftn-worker-μ9bp	Ready	worker	26	3.51 GiB / 15.57 GiB	0.376 cores / 4 cores	28.41 GiB / 119.8 GiB	Apr 1, 2023, 11:44 PM	-
okdnet-97ftn-worker-kpjt!	Ready	worker	15	1.22 GiB / 15.57 GiB	0.141 cores / 4 cores	11.37 GiB / 119.8 GiB	Apr 1, 2023, 11:47 PM	-
okdnet-97ftn-worker-mz8pm	Ready	worker	15	1.36 GiB / 30.59 GiB	0.146 cores / 8 cores	9.06 GiB / 119.8 GiB	Jun 24, 2023, 1:08 PM	-

Στιγμιότυπο 74. Κάθετη επέκταση υποδομής (2)

4.4.3.6 Αναβάθμιση υποδομής

Η αναβάθμιση της υποδομής OKD σχετίζεται με την αναβάθμιση της ίδιας της πλατφόρμας αλλά και της υποδομής οVirt, στην οποία είναι εγκατεστημένη. Για την οποιαδήποτε αναβάθμιση ενός εκ των δύο υποδομών απαιτείται ο έλεγχος της συμβατότητας των εκδόσεων τους, σύμφωνα με τον πίνακα συμβατότητας που παρέχεται από τον προμηθευτή (Red Hat) του λογισμικού.

	RHV 4.3	RHV 4.4
OCP 4.4	yes	yes (tested on RHV 4.4.2 or later)
OCP 4.5	yes	yes (tested on RHV 4.4.2 or later)
OCP 4.6	no (no update possible)	yes (tested on RHV 4.4.4 or later)
OCP 4.7	no	yes (tested on RHV 4.4.6 or later)
OCP 4.8	no	yes (tested on RHV 4.4.6 or later)
OCP 4.9	no	yes (tested on RHV 4.4.6 or later)
OCP 4.10	no	yes (tested on RHV 4.4.6 or later)

Εικόνα 5. Πίνακας συμβατότητας εκδόσεων OKD και RHV

Αξίζει να σημειωθεί ότι δεν διατίθεται στην επίσημη τεκμηρίωση πίνακας συμβατότητας για εκδόσεις νεότερες της πλατφόρμας RHV 4.4 (τρέχουσα oVirt έκδοση 4.5.4). Πέραν της συμβατότητας που παρατίθεται από τον πίνακα συμβατότητας και εξασφαλίζει την ομαλή λειτουργία των πλατφορμών, κατά τη διαδικασία αναβάθμισης σημαντικός παράγοντας που πρέπει να συνυπολογίζεται είναι και η συμβατότητα αναβάθμισης μεταξύ των εκδόσεων του ίδιου του λογισμικού (π.χ. αναβάθμιση από έκδοση 4.1 σε 4.12, προϋποθέτει αναβαθμίσεις σε ενδιάμεσες εκδόσεις).

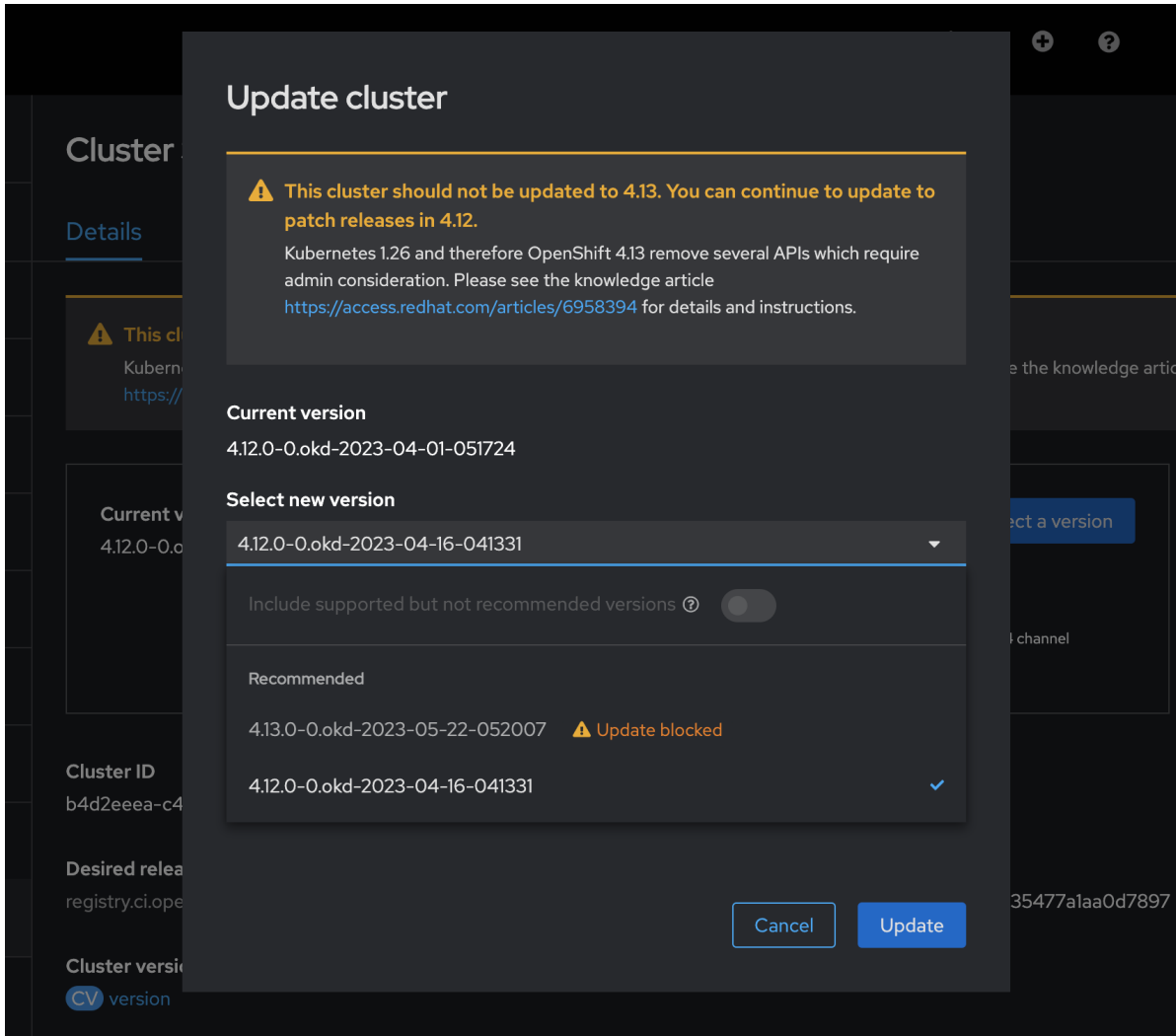
Η αναβάθμιση του oVirt σύμφωνα με την επίσημη τεκμηρίωση εμπεριέχει δύο στάδια. Το πρώτο στάδιο αφορά την αναβάθμιση του oVirt engine (λογισμικό διαχείρισης της συστοιχίας) και το δεύτερο στάδιο την αναβάθμιση των κόμβων που αποτελούν την συστοιχία. Η διαδικασία αναβάθμισης του oVirt engine πραγματοποιείται μέσω της αναβάθμισης των αποθετηρίων του oVirt, των αντίστοιχων πακέτων λογισμικού και την εκτέλεση αρχείου γραμμής εντολών (script) αναβάθμισης. Η διαδικασία αναβάθμισης των κόμβων της συστοιχίας μπορεί να επιτευχθεί είτε μέσω του γραφικού περιβάλλοντος που διατίθεται από το oVirt engine, είτε μέσω αναβάθμισης των πακέτων λογισμικού με την χρήση αποθετηρίων και γραμμής εντολών. Η ομάδα διαχείρισης συστημάτων και δικτύων αξιοποιώντας την μακρόχρονη εμπειρία που διαθέτει στην πλατφόρμα oVirt ακολουθεί την δεύτερη μέθοδο, δεδομένου ότι αφενός έχει ολοκληρωθεί επιτυχώς

σε σειρά αναβαθμίσεων και αφετέρου η διαδικασία αναβάθμισης μέσω του γραφικού περιβάλλοντος δεν ήταν διαθέσιμη σε προγενέστερες εκδόσεις.

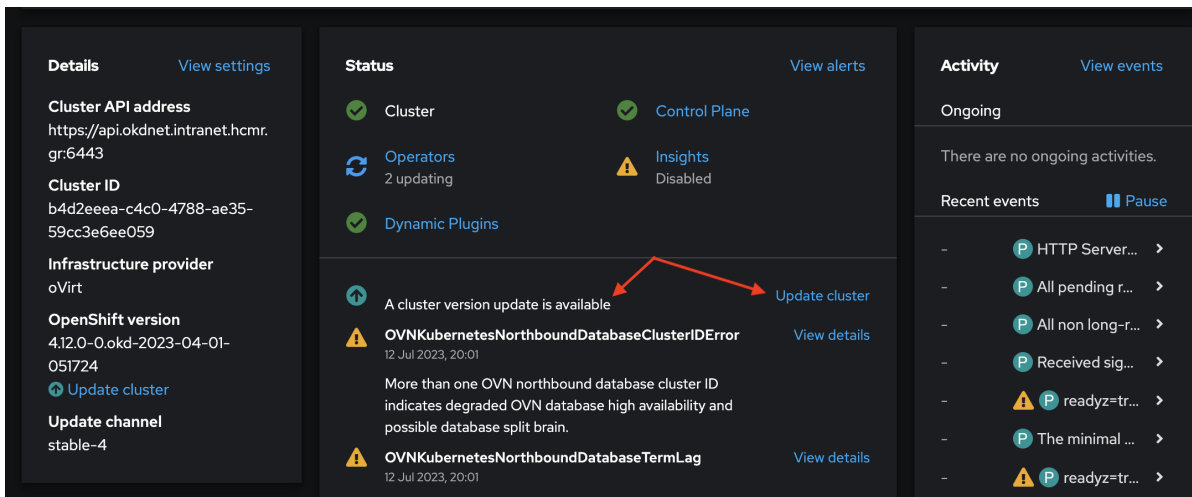
Η αναβάθμιση της πλατφόρμας OKD, σύμφωνα με την επίσημη τεκμηρίωση, επιτυγχάνεται είτε μέσω του γραφικού περιβάλλοντος (OpenShift web console), είτε μέσω της γραμμής εντολών (OpenShift CLI - oc). Και οι δύο μέθοδοι υποστηρίζουν αναβαθμίσεις μεταξύ major και minor εκδόσεων, ενώ παράλληλα πραγματοποιούν τους απαραίτητους ελέγχους αναφορικά με την συμβατότητα μεταξύ των εκδόσεων και παρέχουν την σχετική πληροφόρηση μέσω ειδοποιήσεων στους διαχειριστές της πλατφόρμας. Η διαδικασία αναβάθμισης μέσω του γραφικού περιβάλλοντος είναι φιλική προς τον χρήστη και αυτοματοποιημένη, αλλά στηρίζεται στους ελέγχους του συστήματος και στην επιτυχή ολοκλήρωση των αυτοματοποιημένων διαδικασιών. Η διαδικασία μέσω της γραμμής εντολών, η οποία τεκμηριώνεται επαρκώς, περιλαμβάνει αρκετά στάδια αλλά και χειροκίνητους ελέγχους, που διασφαλίζουν την επιτυχή ολοκλήρωση κάθε βήματος. Παρόλα αυτά, η διαδικασία μέσω της γραμμής εντολών απαιτεί εξοικείωση και εμπειρία με τη χρήση αυτής. Η ομάδα διαχείρισης συστημάτων και δικτύων στα πλαίσια δοκιμών, αναβάθμισε την πλατφόρμα OKD μεταξύ minor εκδόσεων με τη χρήση του γραφικού περιβάλλοντος.

The screenshot displays the 'Cluster Settings' interface. At the top, there are tabs for 'Details', 'ClusterOperators', and 'Configuration'. A warning message states: 'This cluster should not be updated to 4.13. You can continue to update to patch releases in 4.12. Kubernetes 1.26 and therefore OpenShift 4.13 remove several APIs which require admin consideration. Please see the knowledge article https://access.redhat.com/articles/6958394 for details and instructions.' Below this, the 'Current version' is 4.12.0-0.okd-2023-04-01-051724. The 'Update status' section shows 'Available updates' with a timeline of versions: 4.12.0-0.okd-2023-04-01-051724, 4.12.0-0.okd-2023-04-16-041331, and 4.13.0-0.okd-2023-05-22-052007. The 'Channel' is 'stable-4' with a 'Select a version' button. The timeline indicates that the current version is at the first point, and the next available update is the second point, with a warning icon above the third point.

Στιγμιότυπο 75. Προβολή διαθέσιμης αναβάθμισης OKD (1)



Στιγμιότυπο 76. Προβολή διαθέσιμης αναβάθμισης OKD (2)



Στιγμιότυπο 77. Προβολή διαθέσιμης αναβάθμισης OKD (3)

```

root@okdnet-97ftn-master-0:/var/home/core:145x30
[root@okdnet-97ftn-master-0 core]# oc adm upgrade
Cluster version is 4.12.0-0.okd-2023-04-01-051724

Upgradeable=False

Reason: AdminAckRequired
Message: Kubernetes 1.26 and therefore OpenShift 4.13 remove several APIs which require admin consideration. Please see the knowledge article h
tps://access.redhat.com/articles/6958394 for details and instructions.

Upstream: https://amd64.origin.releases.ci.openshift.org/graph
Channel: stable-4

Recommended updates:

VERSION          IMAGE
4.13.0-0.okd-2023-05-22-052007 registry.ci.openshift.org/origin/release@sha256:16b7227d0c0de198d65c4478617e402d7f67ffc0dee312110a9d4c85877e5e4d
4.12.0-0.okd-2023-04-16-041331 registry.ci.openshift.org/origin/release@sha256:2b3d90157565bb1e227c1cd182154b498c4cf76360d8a57cc5d6d5a4a63794cb
[root@okdnet-97ftn-master-0 core]#

```

Στιγμιότυπο 78. Προβολή διαθέσιμης αναβάθμισης OKD μέσω γραμμής εντολών

4.4.3.7 Έλεγχος για υψηλή διαθεσιμότητα

Οι δοκιμές υψηλής διαθεσιμότητας προσομοιώνουν τις καταστάσεις αυτές για να εξακριβώσουν την ικανότητα της υποδομής να αντιμετωπίσει τέτοιες προκλήσεις και να εξασφαλίσει συνεχή λειτουργία και διαθεσιμότητα των υπηρεσιών. Οι δοκιμές υψηλής διαθεσιμότητας αποτελούν ζωτικό βήμα για την αξιολόγηση της απόδοσης και της αξιοπιστίας της υποδομής, καθώς και για την αναγνώριση πιθανών σημείων αστοχίας και την λήψη απαραίτητων μέτρων για τη βελτίωση της διαθεσιμότητας. Στην προκειμένη υποδομή η ομάδα διαχείρισης συστημάτων και δικτύων εκτέλεσε δοκιμές βασιζόμενες σε τρία διαφορετικά σενάρια με σκοπό την κάλυψη των συνηθέστερων περιπτώσεων αστοχίας ή αποτυχίας της υποδομής.

Το **πρώτο σενάριο** αφορά την αστοχία της υποδομής οVirt και πιο συγκεκριμένα την απώλεια κόμβου της συστοιχίας στην οποία φιλοξενείται η πλατφόρμα OKD. Λόγω του ότι η πλατφόρμα οVirt χρησιμοποιείται από τον Φορέα ως IaaS υποδομή, επί σειρά ετών, έχουν πραγματοποιηθεί σχετικοί έλεγχοι είτε μέσω δοκιμών, είτε λόγω πραγματικής αστοχίας κόμβου στα πλαίσια της παραγωγικής λειτουργίας της υποδομής. Για λόγους πολιτικής ορθότητας, εκπονήθηκε σχετική δοκιμή η οποία στέφθηκε με επιτυχία και οι εικονικές μηχανές που υποστήριζαν την πλατφόρμα OKD, επανεκκινήθηκαν στους υπόλοιπους διαθέσιμους κόμβους.

Τα **άλλα δύο σενάρια** περιλάμβαναν δοκιμές σχετικά με την αστοχία / απώλεια master ή worker κόμβων. Σύμφωνα με τις προδιαγραφές του OKD και του Kubernetes στο οποίο βασίζεται, οι master κόμβοι οι οποίοι αξιοποιούν την βάση δεδομένων etcd, για την αποτύπωση της κατάστασης της συστοιχίας, παρέχουν ανοχή σε σφάλματα με βάση τον τύπο $(n/2)+1$, όπου n είναι ο αριθμός των (etcd) κόμβων και το αποτέλεσμα αυτής της πράξης δείχνει πόσοι etcd κόμβοι (majority) απαιτείται να συμφωνούν σε κάποια αλλαγή, ώστε τα δεδομένα της κατάστασης της

συστοιχίας να είναι διασφαλισμένα. Γνωρίζοντας πόσοι είναι οι απαραίτητοι, σε συμφωνία, κόμβοι (majority) για την διασφάλιση της συστοιχίας, είναι δυνατόν με μία επιπλέον αφαίρεση από τους συνολικούς κόμβους να βρεθεί ο αριθμός των κόμβων που μπορούν να αστοχήσουν. Αυτό οφείλεται στον αλγόριθμο ομοφωνίας Raft (consensus algorithm) [71] που ενσωματώνει το λογισμικό etcd και ο οποίος απαιτεί πλειοψηφία $50\% + 1$ κόμβων για να εξασφαλιζεται συνέπεια των δεδομένων.

Cluster Size	Majority	Failure Tolerance
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3
8	5	3
9	5	4

Εικόνα 6. Πίνακας κόμβων etcd με βάση τον αλγόριθμο ομοφωνίας Raft

Σύμφωνα με την υποδομή που έχει υλοποιηθεί και αποτελείται από τρεις master κόμβους οι οποίοι εμπεριέχουν την βάση etcd, το περιθώριο αστοχίας περιορίζεται στην απώλεια ενός κόμβου. Στα πλαίσια της δοκιμής απενεργοποιήθηκε χειροκίνητα ένας εκ των τριών master κόμβων και εν συνεχεία ελέγχθηκε η ορθή λειτουργία της πλατφόρμας OKD, μετά την απώλεια του κόμβου αυτού.

Name	Comment	Uptime	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics
okdnet-97ftn-master-0		1 day	10.5.1.125 10.0.40.87 169.254.169.2 fe80::46a7:91...	vmcloud74	AnavissosDC	38%	35%	0%	None
okdnet-97ftn-master-1		1 day	10.5.1.104 169.254.169.2 fe80::984d:bcf2:f920:e55e	vmcloud73	AnavissosDC	48%	50%	0%	None
okdnet-97ftn-master-2					AnavissosDC	--	--	--	None
okdnet-97ftn-worker-87d8q		12 min	10.5.1.105 169.254.169.2 fe80::2442:769:388f:f459	vmcloud71	AnavissosDC	16%	10%	0%	VNC
okdnet-97ftn-worker-jx9bp		12 min	10.5.1.134 169.254.169.2 10.0.40.88 fe80::80b3:23...	vmcloud72	AnavissosDC	18%	15%	0%	VNC
okdnet-97ftn-worker-kpjtl		1 day	10.5.1.171 169.254.169.2 10.0.40.88 fe80::cd46:a3...	vmcloud73	AnavissosDC	15%	7%	0%	VNC
okdnet-97ftn-worker-mz8pm		1 day	10.5.1.156 169.254.169.2 fe80::8fd7:85bc:ebe0:dca	vmcloud74	AnavissosDC	5%	2%	0%	VNC
okdprovcloud		1 day	10.5.1.109 fe80::546f:83ff:feb3:a	vmcloud72	AnavissosDC	14%	0%	0%	VNC

Στιγμιότυπο 79. Έλεγχος υψηλής διαθεσιμότητας OKD (1)

Nodes

Filter Name Search by name... /

Name	Status	Roles	Pods	Memory	CPU	Filesystem	Created	Instance type
okdhnet-97ftn-master-0	Ready	control-plane, master	45	6.03 GiB / 15.57 GiB	0.827 cores / 4 cores	57.47 GiB / 119.8 GiB	1 Apr 2023, 22:38	-
okdhnet-97ftn-master-1	Ready	control-plane, master	67	7.58 GiB / 15.57 GiB	0.996 cores / 4 cores	25.75 GiB / 119.8 GiB	1 Apr 2023, 22:36	-
okdhnet-97ftn-master-2	Not Ready	control-plane, master	-	-	-	-	1 Apr 2023, 22:38	-

Στιγμιότυπο 80. Έλεγχος υψηλής διαθεσιμότητας OKD (2)

Control Plane status

Components of the Control Plane are responsible for maintaining and reconciling the state of the cluster.

Components	Response rate
API Servers	100% ✓
Controller Managers	66.7% !
Schedulers	66.7% !
API Request Success Rate	100% ✓

Status

Cluster ✓

Control Plane !
2 components degraded

Στιγμιότυπο 81. Έλεγχος υψηλής διαθεσιμότητας OKD (3)

etcd ⚠ Degraded 4.12.0-0.okd-2023-04-01-051724

ClusterMemberControllerDegraded: unhealthy members found during reconciling members
EtcdEndpointsDegraded: EtcdEndpointsController can't evaluate whether quorum is safe: etcd cluster has quorum of 2 and 2 healthy members which is not fault tolerant:

Στιγμιότυπο 82. Έλεγχος υψηλής διαθεσιμότητας OKD (4)

Αναφορικά με την απώλεια worker κόμβων πρέπει να σημειωθεί ότι δεν υπάρχει σταθερός τύπος για τον υπολογισμό του ακριβή αριθμού κόμβων που μπορούν να

αστοχήσουν χωρίς να επηρεαστεί η συστοιχία. Ο υπολογισμός εξαρτάται από τα συγκεκριμένα χαρακτηριστικά των εφαρμογών, τον φόρτο εργασίας τους και τους διαθέσιμους πόρους των κόμβων που τις φιλοξενούν. Λαμβάνοντας υπόψη αυτούς τους παράγοντες και διενεργώντας κατάλληλες δοκιμές, μπορεί να προσδιοριστεί ο κατάλληλος αριθμός των worker κόμβων και να υλοποιηθούν κατάλληλες πολιτικές για την διασφάλιση της υψηλής διαθεσιμότητας και την ανεκτικότητα στα σφάλματα της συστοιχίας OKD. Στην συγκεκριμένη περίπτωση και λόγω του μειωμένου φόρτου εργασίας, πραγματοποιήθηκε δοκιμή κατά την οποία απενεργοποιήθηκαν δύο από τους τέσσερις worker κόμβους. Στη συνέχεια έγιναν οι απαραίτητοι έλεγχοι ορθής λειτουργίας των εφαρμογών, επιβεβαιώνοντας τις δυνατότητες της πλατφόρμας για παροχή υπηρεσιών υψηλής διαθεσιμότητας. Πιο αναλυτικά, επιλέχθηκε η απενεργοποίηση worker κόμβου, στον οποίο φιλοξενούνταν εφαρμογή δικτυακού εξυπηρετητή (nginx), επιβεβαιώθηκε η επαναφορά της εφαρμογής σε άλλο διαθέσιμο worker κόμβο και παρατηρήθηκαν οι χρόνοι απόκρισης και οι αυτοματοποιημένες ενέργειες της συστοιχίας. Στην συνέχεια, ακολούθησε απενεργοποίηση και δεύτερου worker κόμβου, σύμφωνα με την σχεδίαση της δοκιμής, και επιβεβαιώθηκε εκ νέου η επαναφορά της εφαρμογής σε έναν από τους δύο εναπομείναντες worker κόμβους.

Pods > Pod details

P nginx-596859499c-2hqpq Running

Details Metrics YAML Environment Logs Events Terminal

Pod details

<p>Name</p> <p>nginx-596859499c-2hqpq</p>	<p>Status</p> <p>Running</p>
<p>Namespace</p> <p>NS nginxiordanism</p>	<p>Restart policy</p> <p>Always restart</p>
<p>Labels Edit</p> <p>app.kubernetes.io/instance=nginx app.kubernetes.io/managed-by=Helm app.kubernetes.io/name=nginx</p> <p>helm.sh/chart=nginx-15.1.1 pod-template-hash=596859499c</p>	<p>Active deadline seconds</p> <p>Not configured</p>
<p>Node selector</p> <p>No selector</p>	<p>Pod IP</p> <p>10.128.2.30</p>
<p>Tolerations</p> <p>2 tolerations</p>	<p>Host IP</p> <p>10.5.1.134</p>
<p>Annotations</p> <p>4 annotations</p>	<p>Node</p> <p>okdnet-97ftn-worker-jx9bp</p>
<p>Created at</p> <p>13 Jul 2023, 18:34</p>	<p>Image pull secret</p> <p>default-dockercfg-cpbqg</p>
<p>Owner</p> <p>RS nginx-596859499c</p>	<p>PodDisruptionBudgets</p> <p>No PodDisruptionBudgets</p>

Στιγμιότυπο 83. Έλεγχος κόμβου εφαρμογής nginx

Compute > Virtual Machines

Vms: okd

Name	Comment	Uptime	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics
okdnet-97ftn-master-0		1 day	10.5.1.125 10.0.40.87 169.254.169.2 fe80::46a7:91...	vmcloud74	AnavissosDC	37%	28%	0%	None
okdnet-97ftn-master-1		1 day	10.5.1.104 169.254.169.2 fe80::984d:bcf2:f920:e55e	vmcloud73	AnavissosDC	46%	29%	0%	None
okdnet-97ftn-master-2		1 day	10.5.1.117 169.254.169.2 fe80::8596:e41b:374f:9725	vmcloud72	AnavissosDC	32%	25%	0%	None
okdnet-97ftn-worker-87d8q		5 min	10.5.1.105 169.254.169.2 fe80::cb73:b832:bb26:4b...	vmcloud71	AnavissosDC	15%	10%	0%	VNC
okdnet-97ftn-worker-jx9bp					AnavissosDC	--	--	--	None
okdnet-97ftn-worker-kpjtl		1 day	10.5.1.171 169.254.169.2 10.0.40.88 fe80::cd46:a3...	vmcloud73	AnavissosDC	15%	6%	0%	VNC
okdnet-97ftn-worker-mz8pm		1 day	10.5.1.156 169.254.169.2 fe80::8fd7:85bc:ebe0:dcda	vmcloud72	AnavissosDC	5%	2%	0%	VNC
okdprovcloud		1 day	10.5.1.109 fe80::546f:83ff:feb3:a	vmcloud72	AnavissosDC	14%	0%	0%	VNC

Στιγμιότυπο 84. Έλεγχος υψηλής διαθεσιμότητας OKD (5)

Pods > Pod details

P nginx-596859499c-f8pvt Running

Details Metrics YAML Environment Logs Events Terminal

Pod details

Name
nginx-596859499c-f8pvt

Status
Running

Namespace
NS nginxordanism

Restart policy
Always restart

Labels Edit

app.kubernetes.io/instance=nginx app.kubernetes.io/managed-by=Helm app.kubernetes.io/name=nginx
helm.sh/chart=nginx-15.1.1 pod-template-hash=596859499c

Active deadline seconds
Not configured

Pod IP
10.131.0.25

Node selector
No selector

Host IP
10.5.1.105

Tolerations
2 tolerations

Node
N okdnet-97ftn-worker-87d8q

Annotations
4 annotations

Image pull secret
S default-dockercfg-cpbqg

Created at
13 Jul 2023, 18:43

PodDisruptionBudgets
No PodDisruptionBudgets

Στιγμιότυπο 85. Έλεγχος επαναφοράς εφαρμογής σε εναπομείναντα κόμβο (1)

Name	Comment	Uptime	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics
okdnet-97ftn-master-0		1 day	10.5.1.125 10.0.40.87 169.254.169.2 fe80::46a791...	vmcloud74	AnavissosDC	38%	25%	0%	None
okdnet-97ftn-master-1		1 day	10.5.1.104 169.254.169.2 fe80::984d:bcf2:f920:e55e	vmcloud73	AnavissosDC	47%	28%	0%	None
okdnet-97ftn-master-2		1 day	10.5.1.117 169.254.169.2 fe80::8596:e41b:374f:9725	vmcloud72	AnavissosDC	32%	21%	0%	None
okdnet-97ftn-worker-87d8q					AnavissosDC	--	--	--	None
okdnet-97ftn-worker-jx9bp					AnavissosDC	--	--	--	None
okdnet-97ftn-worker-kpjtj		1 day	10.5.1.171 169.254.169.2 10.0.40.88 fe80::cd46:a3...	vmcloud73	AnavissosDC	15%	15%	0%	VNC
okdnet-97ftn-worker-mz8pm		1 day	10.5.1.156 169.254.169.2 fe80::8fd7:85bc:ebe0:dcd4	vmcloud72	AnavissosDC	5%	2%	0%	VNC
okdprovcloud		1 day	10.5.1.109 fe80::546f:83ff:feb3:a	vmcloud72	AnavissosDC	14%	0%	0%	VNC

Στιγμιότυπο 86. Έλεγχος υψηλής διαθεσιμότητας OKD (6)

Pods > Pod details

P nginx-596859499c-71q22 Running

Details Metrics YAML Environment Logs Events Terminal

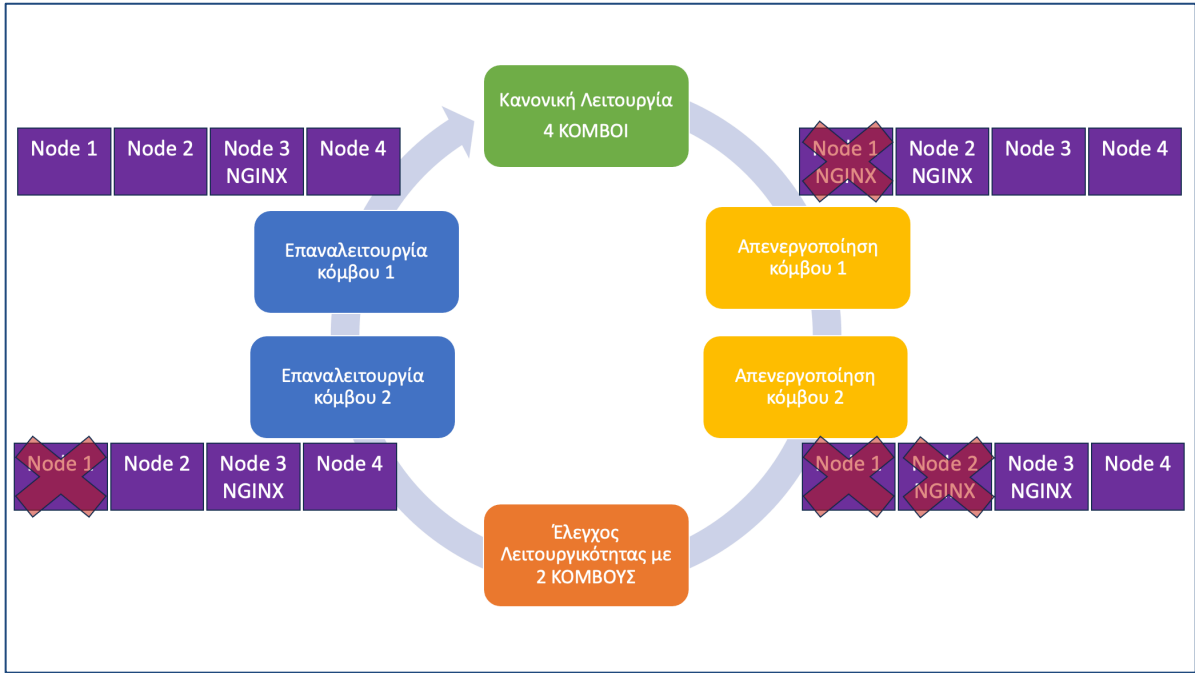
Pod details

Name nginx-596859499c-71q22	Status Running
Namespace NS nginxordanism	Restart policy Always restart
Labels <div style="border: 1px solid #ccc; padding: 2px; display: flex; gap: 5px;"> app.kubernetes.io/instance=nginx app.kubernetes.io/managed-by=Helm app.kubernetes.io/name=nginx helm.sh/chart=nginx-15.1.1 pod-template-hash=596859499c </div>	Active deadline seconds Not configured
Node selector No selector	Pod IP 10.129.2.101
Tolerations 2 tolerations	Host IP 10.5.1.171
Annotations 4 annotations	Node N okdnet-97ftn-worker-kpjtl
Created at 🕒 13 Jul 2023, 18:52	Image pull secret S default-dockercfg-cpbqg
Owner RS nginx-596859499c	PodDisruptionBudgets No PodDisruptionBudgets

Στιγμιότυπο 87. Έλεγχος επαναφοράς εφαρμογής σε εναπομείναντα κόμβο (2)

P nginx-596859499c-2hpgp	Terminating	1/1	0	RS nginx-596859499c	-	-	🕒 13 Jul 2023, 18:34	⋮
P nginx-596859499c-71q22	Running	1/1	0	RS nginx-596859499c	-	-	🕒 13 Jul 2023, 18:52	⋮
P nginx-596859499c-f8pvt	Terminating	1/1	0	RS nginx-596859499c	-	-	🕒 13 Jul 2023, 18:43	⋮

Στιγμιότυπο 88. Έλεγχος υψηλής διαθεσιμότητας OKD (7)



Σχήμα 11. Σχηματική απεικόνιση διαδικασίας ελέγχου υψηλής διαθεσιμότητας OKD

4.4.3.8 Δοκιμή ολικής απενεργοποίησης και επαναφοράς

Όταν μια υποδομή τίθεται εκτός λειτουργίας, είτε λόγω συντήρησης, είτε λόγω προβλήματος, η διάθεση διαδικασίας για την επαναφορά της σε κανονική λειτουργία αποτελεί αδιαμφισβήτητη απαίτηση. Κατά την επαναφορά, πρέπει να ληφθούν υπόψη διάφοροι παράγοντες, όπως η ασφάλεια των δεδομένων, η συνέπεια του συστήματος και η ελαχιστοποίηση του χρόνου ανακτήσεως. Οι διαδικασίες για την επαναφορά πρέπει να είναι καλά τεκμηριωμένες και να έχουν δοκιμαστεί πριν η υποδομή τεθεί σε παραγωγική λειτουργία. Επιπλέον, η παρακολούθηση της διαδικασίας επαναφοράς είναι εξίσου σημαντική για να διασφαλιστεί ότι η υποδομή επανέρχεται στην κανονική της κατάσταση με επαρκή επίπεδα απόδοσης και λειτουργικότητας.

Σε αντιστοιχία με την δοκιμή που υλοποιήθηκε στην υποδομή Harvester και Rancher (βλ. 4.4.2.8), η ομάδα διαχείρισης συστημάτων και δικτύων, εκπόνησε δοκιμές οι οποίες περιλάμβαναν την ολική απενεργοποίηση της υποδομής OKD και εν συνεχεία την ενεργοποίηση της σύμφωνα με τις διαδικασίες που περιγράφονται στην επίσημη τεκμηρίωση, ώστε να ελεγχθεί η ορθή και πλήρης επαναφορά της λειτουργίας της υποδομής OKD. Πρέπει να σημειωθεί ότι το OKD διαθέτει

τεκμηρίωση και για την ομαλή (graceful) απενεργοποίηση της συστοιχίας και για την ομαλή επανεκκίνηση της συστοιχίας, όπως επίσης διαθέτει και οδηγούς αντιμετώπισης προβλημάτων για τις προαναφερθείσες διαδικασίες [72].

Οι δοκιμές μπορούν να χαρακτηριστούν επιτυχείς, αφού, επί το πλείστον, η συστοιχία επανερχόταν σε πλήρη λειτουργία και η πρόσβαση στις εκτελούμενες εφαρμογές πραγματοποιούνταν όπως αναμενόταν. Σε μία εκ των δοκιμών, παρουσιάστηκε αδυναμία ενσωμάτωσης όλων των master κόμβων, αλλά βάσει των οδηγιών αντιμετώπισης προβλημάτων η ομάδα διαχείρισης συστημάτων και δικτύων επανάφερε την OKD συστοιχία σε ορθή λειτουργία, με ενέργειες που αφορούσαν την αποδοχή αιτήσεων πιστοποιητικών (CSRs).

Compute > Virtual Machines

Vms: okd

Name	Comment	Uptime	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics	Status
okdnet-97ftn-master-0					AnavissosDC	--	--	--	None	Down
okdnet-97ftn-master-1					AnavissosDC	--	--	--	None	Down
okdnet-97ftn-master-2					AnavissosDC	--	--	--	None	Down
okdnet-97ftn-worker-87d8q					AnavissosDC	--	--	--	None	Down
okdnet-97ftn-worker-jx9bp					AnavissosDC	--	--	--	None	Down
okdnet-97ftn-worker-kpjtl					AnavissosDC	--	--	--	None	Down
okdnet-97ftn-worker-mz8pm					AnavissosDC	--	--	--	None	Down

Στιγμιότυπο 89. Δοκιμή ολικής απενεργοποίησης και επαναφοράς (1)

Compute > Virtual Machines

Vms: okd

Name	Comment	Uptime	IP Addresses	Host	Data Center	Memory	CPU	Network	Graphics	Status
okdnet-97ftn-master-0		5 min	10.5.1.125 169.254.169.2 fe80::a302:4c93:31...	vmcloud74	AnavissosDC	36%	24%	0%	None	Up
okdnet-97ftn-master-1		6 min	10.5.1.104 10.0.40.87 169.254.169.2 fe80::dc...	vmcloud73	AnavissosDC	43%	43%	0%	None	Up
okdnet-97ftn-master-2		5 min	10.5.1.117 169.254.169.2 fe80::fd37:fc0fab2...	vmcloud72	AnavissosDC	25%	21%	0%	None	Up
okdnet-97ftn-worker-87d8q		5 min	10.5.1.105 169.254.169.2 fe80::dd93:30d1:f...	vmcloud74	AnavissosDC	17%	17%	0%	VNC	Up
okdnet-97ftn-worker-jx9bp		6 min	10.5.1.134 169.254.169.2 fe80::6f5e:f0d8:15...	vmcloud73	AnavissosDC	18%	20%	0%	VNC	Up
okdnet-97ftn-worker-kpjtl		6 min	10.5.1.171 169.254.169.2 fe80::e07:3727:24...	vmcloud71	AnavissosDC	12%	7%	0%	VNC	Up
okdnet-97ftn-worker-mz8pm		5 min	10.5.1.156 10.0.40.88 169.254.169.2 fe80::9...	vmcloud72	AnavissosDC	4%	2%	0%	VNC	Up

Στιγμιότυπο 90. Δοκιμή ολικής απενεργοποίησης και επαναφοράς (2)

4.5 Συμπεράσματα και Αξιολόγηση λύσεων

Η μελέτη της υπάρχουσας κατάστασης στο Ελληνικό Κέντρο Θαλάσσιων Ερευνών, η ανάλυση των απαιτήσεων, η εκτίμηση των διαθέσιμων λύσεων και η τεχνική αξιολόγηση τους, σε συνδυασμό με την δοκιμή των επικρατέστερων λύσεων, δημιούργησε μια ολοκληρωμένη εικόνα αναφορικά με τις επιλογές, τις δυνατότητες, τις απαιτήσεις σε τεχνογνωσία αλλά και σε πόρους, ώστε να

υλοποιηθεί μια ενιαία υποδομή διαχείρισης εικονικών μηχανών και ενορχήστρωσης περιεκτών. Όλες οι λύσεις ακολουθούν την αρχιτεκτονική επιπέδων (PaaS / CaaS over IaaS), η οποία δεν δεσμεύει ανεξάρτητους φυσικούς πόρους για το κάθε μοντέλο διανομής υπηρεσιών, αλλά παρέχει ευελιξία διαχείρισης φυσικών πόρων, ειδικά σε περιβάλλοντα περιορισμένης διαθεσιμότητας, όπως σε μικρομεσαίες επιχειρήσεις και οργανισμούς. Οι δύο λύσεις που δοκιμάστηκαν και διαθέτουν δυνατότητες διαλειτουργικότητας με IaaS υποδομές, βελτιώνουν περαιτέρω την ευελιξία της αρχιτεκτονικής αυτής, αποδεσμεύοντας τους οργανισμούς από την ανάγκη υπέρμετρης προμήθειας φυσικών πόρων ή εγκατάστασης πιο πολύπλοκων λογισμικών (π.χ. OpenStack) που διαθέτουν εγγενώς την υποστήριξη και των δύο μοντέλων υπηρεσιών (IaaS - PaaS). Επισημαίνεται πως οι επιλογές αυτές δεν αποτελούν γενικευμένη βέλτιστη πρακτική, αλλά στοχεύουν στην αποτελεσματικότερη και στην πληρέστερη αξιοποίηση των περιορισμένων πόρων – συγκριτικά με τα μεγάλα Κέντρα Δεδομένων – των μικρομεσαίων οργανισμών και επιχειρήσεων και αποτελούν εκτιμήσεις της ομάδας διαχείρισης συστημάτων και δικτύων του Ελληνικού Κέντρου Θαλάσσιων Ερευνών.

Και οι δύο τεχνολογικές λύσεις, οι οποίες προκρίθηκαν και δοκιμάστηκαν, εγκαταστάθηκαν επιτυχώς, ανταποκρίθηκαν ικανοποιητικά αναφορικά με τις κύριες λειτουργικές απαιτήσεις και μπορούν να χαρακτηριστούν ως βιώσιμες και επαρκείς για τις ανάγκες του φορέα. Η κάθε λύση παρουσιάζει διαφορετικά προτερήματα και αδυναμίες, με τη λύση του OKD να επικρατεί σε επίπεδο ωριμότητας, λειτουργιών και συμβατότητας με την υφιστάμενη υποδομή (oVirt).

Αξιολογώντας την πλατφόρμα Rancher, πρέπει να τονιστεί πως προσφέρει μια ιδιαίτερα σαφή και φιλική διεπαφή χρήστη, που παρέχει πρόσβαση στην πληθώρα των λειτουργιών της πλατφόρμας και ανταποκρίνεται επαρκώς στις απαιτήσεις του φορέα. Επιπρόσθετα, η πλατφόρμα Rancher υποστηρίζει την διαχείριση πολλαπλών και διαφορετικών συστοιχιών Kubernetes, ενώ διαθέτει πλήρη διεπαφή για τη διάδραση του με το Harvester. Από την άλλη πλευρά, η διαδικασία εγκατάστασης του Rancher προϋποθέτει την ύπαρξη συστοιχίας Kubernetes, καθιστώντας την μη αυτοματοποιημένη, συγκριτικά με την διαδικασία εγκατάστασης του OKD, που επιτυγχάνεται με την εκτέλεση ενός κατάλληλα διαμορφωμένου αρχείου γραμμής εντολών (script). Η πλατφόρμα Harvester, που αποτελεί την IaaS υποδομή της λύσης, προσφέρει εύκολη διαδικασία εγκατάστασης και δημιουργίας

συστοιχίας, ενώ διαθέτει και ένα αρκετά πλήρες και φιλικό περιβάλλον χρήστη. Η πλατφόρμα, αυτή, στηρίζεται στην φιλοσοφία των συστοιχιών Kubernetes και αντιμετωπίζει τις εικονικές μηχανές ως φόρτους εργασίας (workloads) και όχι στην κλασσική φιλοσοφία των υπερεποπτών, που αξιοποιούν τεχνολογίες όπως το KVM, το XEN κ.α. Αν και επικαλείται τη διάθεση πληθώρας δυνατοτήτων, αντίστοιχων των διαχειριστών εικονικών μηχανών (VMMs) στερείται ωριμότητας και αξιοπιστίας, καθώς βασίζεται υπό ανάπτυξη λογισμικά όπως το KubeVirt, που κατά τη διάρκεια των δοκιμών διατίθετο στην έκδοση 0.54. Το γεγονός αυτό, καθιστά την πλατφόρμα Harvester, επιρρεπή σε σφάλματα σχετικά με πιο εξειδικευμένες λειτουργίες εικονικών μηχανών, όπως η διαχείριση στιγμιότυπων (snapshots), κλώνων και αντιγράφων ασφαλείας αλλά και λειτουργιών υψηλής διαθεσιμότητας όπως η μετακίνηση εικονικών μηχανών εν ώρα λειτουργίας (live migration). Πρέπει, όμως, να σημειωθεί πως λόγω της συνεχούς ανάπτυξης και εξέλιξης τους παραμένουν πολλά υποσχόμενες ως τεχνολογίες.

Η εταιρός πλατφόρμα που υλοποιήθηκε, το OKD, όπως προαναφέρθηκε εγκαθίσταται μέσω της εκτέλεσης ενός διαμορφώσιμου αρχείου γραμμής εντολών (script), καθιστώντας την διαδικασία αυτοματοποιημένη. Η ορθή και πλήρης παραμετροποίηση προϋποθέτει ολοκληρωμένη κατανόηση της αρχιτεκτονικής του OKD, όπως και την παραμετροποίηση των υποδομών δικτύου (DNS, DHCP) και των υποδομών IaaS (oVirt – template images). Η διεπαφή του OKD, αν και πλήρης, ίσως απαιτεί μεγαλύτερο επίπεδο εξοικείωσης αφού διαχωρίζεται σε δύο διεπαφές, την κονσόλα του Διαχειριστή και του Προγραμματιστή. Το OKD, υποστηρίζει μία συστοιχία Kubernetes σε αντίθεση με το Rancher, που μπορεί να υποστηρίξει πολλές διαφορετικές, αλλά προσφέρει τη δυνατότητα δημιουργίας συνόλων μηχανών (machine sets) που μπορούν να ανατεθούν σε διαφορετικά είδη εργασιών και χρηστών. Η εγκατάσταση της IaaS υποδομής (oVirt) μπορεί να θεωρηθεί πιο περίπλοκη συγκριτικά με την εγκατάσταση του Harvester, αλλά διαθέτει αυξημένες λειτουργίες, ενώ παράλληλα η ομάδα διαχείρισης συστημάτων και δικτύων διαθέτει αρκετή εμπειρία και τεχνογνωσία, αφού αποτελεί την παραγωγική IaaS λύση του φορέα.

Και οι δύο λύσεις καλύπτουν επαρκώς τις λειτουργικές απαιτήσεις αναφορικά με την διαχείριση περιεκτών και γενικότερα της διάθεσης πλατφόρμας διαχείρισης περιεκτών (PaaS) σε χρήστες, διασφαλίζοντας λειτουργική και δικτυακή

απομόνωση. Παράλληλα διαθέτουν τις απαιτούμενες διεπαφές προς τους διαχειριστές του συστήματος ώστε να διατηρούν μια καθολική εικόνα της υποδομής. Η εγκατάσταση των δοκιμαστικών εφαρμογών μέσω περιεκτών ήταν επιτυχής και στις δύο πλατφόρμες με μικρές διαφορές ως προς την διαδικασία υλοποίησης, ενώ και οι δύο πλατφόρμες παρέχουν εύκολη πρόσβαση σε ιδιωτικά και δημόσια αποθετήρια, όπως και στη συνολικότερη διαχείριση των πόρων που μπορεί να αιτηθεί ο εκάστοτε χρήστης. Η πλατφόρμα Rancher μπορεί να θεωρηθεί ότι παρέχει μεγαλύτερη απομόνωση, αφού προσφέρει τη δυνατότητα διάθεσης ανεξάρτητης συστοιχίας Kubernetes ανά χρήστη / ομάδα εργασίας, δεσμεύοντας ταυτόχρονα περισσότερους πόρους συγκριτικά με τη πλατφόρμα OKD, που ακόμα και στην περίπτωση διάθεσης ξεχωριστού συνόλου από worker κόμβους (machine set) δεν απαιτεί επιπρόσθετους master κόμβους. Και οι δύο λύσεις προσφέρουν δυνατότητες για στιγμιότυπα και κλώνους εικονικών μηχανών, όπως και διαδικασίες αντιγράφων ασφαλείας.

Συνοψίζοντας, οι τεχνολογικές λύσεις που επιλέχθηκαν και δοκιμάστηκαν από την ομάδα διαχείρισης συστημάτων και δικτύων, του ΕΛΚΕΘΕ, κατέδειξαν την δυνατότητα υποστήριξης φιλοξενίας πλατφόρμας ενορχήστρωσης περιεκτών και εικονικών μηχανών στην ίδια υποδομή και υπό περιορισμένους φυσικούς πόρους. Προκύπτει αβίαστα το συμπέρασμα ότι η εγκατάσταση μιας τέτοιας τεχνολογικής λύσης και η ενσωμάτωση της στις διαδικασίες του φορέα, απαιτεί την κατάρτιση του προσωπικού για την απόκτηση της απαιτούμενης τεχνογνωσίας αλλά και ένα εύλογο χρονικό περιθώριο για την πλήρη αφομοίωση της. Σύμφωνα με τις επιλογές της ομάδας, η πλατφόρμα OKD, σε συνδυασμό με το oVirt, επικρατεί της πλατφόρμας Rancher, σε συνδυασμό με το Harvester. Καταλυτικός παράγοντας για την επιλογή αυτή, αποτέλεσε η αξιοπιστία και η σταθερότητα της υποδομής oVirt έναντι της υποδομής Harvester, η οποία κατά την περίοδο των δοκιμών παρουσίασε σφάλματα, όπως και το γεγονός της ωριμότητας του λογισμικού της πλατφόρμας oVirt έναντι της πλατφόρμας Harvester. Σε κάθε περίπτωση, οι δύο λύσεις μπορούν να θεωρηθούν ανταγωνιστικές, και λαμβάνοντας υπόψιν την συνεχή εξέλιξη στο χώρο των νεφοϋπολογιστικών υποδομών και την ανάδειξη νέων λύσεων (π.χ. TrueNAS Scale), δεν μπορεί να αποκλειστεί η μετάβαση σε εναλλακτική πλατφόρμα στο μέλλον.

ΚΕΦΑΛΑΙΟ 5

Συμπεράσματα και Προοπτικές

Στο τελευταίο κεφάλαιο, της παρούσας διπλωματικής, θα επανεξεταστούν οι στόχοι της έρευνας, θα συζητηθούν τα κύρια ευρήματα και θα μελετηθεί η συμβολή της έρευνας στον τομέα των ιδιωτικών νεφοϋπολογιστικών υποδομών. Θα γίνει αναφορά στους περιορισμούς της έρευνας και θα καταγραφούν προτάσεις για μελλοντική έρευνα, σύμφωνα με το ερευνητικό κενό που προέκυψε.

5.1 Ανασκόπηση Στόχων και Ανάλυση Ευρημάτων

Οι αρχικοί στόχοι της παρούσας έρευνας περιλάμβαναν την κατανόηση, μέσω της βιβλιογραφικής ανασκόπησης και της μελέτης περίπτωσης, της επίδρασης της χρήσης των περιεκτών σε ιδιωτικές νεφοϋπολογιστικές υποδομές. Ταυτόχρονα, παρουσιάζεται και αναλύεται η ανάγκη για μετασχηματισμό των υποδομών, με σκοπό την ενσωμάτωση περιεκτών σε παραλληλία με υποδομές εικονικών μηχανών, εξετάζονται πιθανές υλοποιήσεις και μελετάται η περίπτωση μετασχηματισμού των υποδομών ενός Ελληνικού Ερευνητικού Κέντρου.

Στα πλαίσια της έρευνας, παρατηρήθηκε ότι η χρήση των περιεκτών και η αναδιαμόρφωση των υποδομών και υπηρεσιών, καθίσταται πλέον αναγκαία, ώστε να μπορούν να ενσωματωθούν με βέλτιστο τρόπο και να αξιοποιηθούν στο έπακρο οι δυνατότητες που προσφέρουν. Η βιβλιογραφική ανασκόπηση, η εκτεταμένη χρήση και διάθεση περιεκτών, σε συνδυασμό με την cloud native αρχιτεκτονική που καθιερώνεται στο χώρο της ανάπτυξης λογισμικού, επιβεβαιώνουν τα ευρήματα αυτά και προωθούν, σημαντικές, αλλαγές στην δομή και στην λειτουργία των ιδιωτικών νεφοϋπολογιστικών υποδομών.

Η μελέτη των τεχνολογιών και των διαθέσιμων εργαλείων / πλατφορμών κατέδειξαν τις μεθοδολογίες για τον μετασχηματισμό και την προσαρμογή των υποδομών, ώστε να αντιμετωπιστούν οι προκλήσεις που προκύπτουν στις, μικρής κλίμακας, ιδιωτικές νεφοϋπολογιστικές υποδομές. Πιο συγκεκριμένα, η βιβλιογραφική ανασκόπηση των τεχνολογιών περιεκτών, εικονικών μηχανών και εικονικοποίησης γενικότερα, βοήθησε στην κατανόηση της λειτουργίας τους, των δυνατοτήτων συνύπαρξής τους σε κοινές υποδομές και στην εξεύρεση

μεθοδολογιών μετασχηματισμού υπηρεσιών και υποδομών, με τρόπο που να αξιοποιούνται κατά το μέγιστο οι διαθέσιμοι υπολογιστικοί πόροι και τα οφέλη της κάθε τεχνολογίας.

Ο τελικός στόχος της έρευνας αφορά την μελέτη μετασχηματισμού, μικρής κλίμακας, ιδιωτικής νεφούπολογιστικής υποδομής ώστε να είναι ικανή να φιλοξενεί παράλληλα και ολοκληρωμένα περιέκτες και εικονικές μηχανές. Το στάδιο της έρευνας αυτό, αξιοποιεί την αποκτηθείσα γνώση, ενώ παράλληλα εξετάζει σε ρεαλιστικό επίπεδο τις επιλογές και τις δυνατότητες που διαθέτει ένας μικρομεσαίος οργανισμός, με ιδιωτικές υποδομές και καθορισμένους υλικούς και ανθρώπινους πόρους. Με την μελέτη περίπτωσης επιβεβαιώνεται η δυνατότητα μετασχηματισμού μικρής κλίμακας υποδομών, ενώ παράλληλα καταγράφονται οι δυνατότητες, οι επιλογές και οι δυσκολίες υλοποίησης ενός τέτοιου εγχειρήματος.

5.2 Περιορισμοί της Έρευνας

Παρά την εκτεταμένη και ενδελεχή μελέτη που εκπονήθηκε κατά τη διάρκεια της παρούσας έρευνας, είναι σημαντικό να αναγνωριστούν ορισμένοι περιορισμοί, που προέκυψαν κατά τη διάρκεια της διαδικασίας. Οι περιορισμοί αυτοί, δεν υπονομεύουν τη σημασία των ευρημάτων, αλλά παρέχουν ένα επιπλέον πλαίσιο για την κατανόηση του οικοσυστήματος εικονικοποίησης υποδομών και της συμβατότητας των διαθέσιμων επιλογών, σύμφωνα με τις ανάγκες του εκάστοτε οργανισμού. Επιπλέον, η αναγνώρισή τους ενδέχεται να ενισχύσει την ακρίβεια και την αξιοπιστία μελλοντικών ερευνών.

Ο πρώτος περιορισμός αφορά τη διαθεσιμότητα επιστημονικής βιβλιογραφίας αναφορικά με τις μικρής κλίμακας υποδομές φιλοξενίας περιεκτών και, κυρίως, την έλλειψη βιβλιογραφίας σχετικής με υποδομές που υποστηρίζουν παράλληλα εικονικές μηχανές και περιέκτες. Λόγω της επικράτησης των μεγάλων και δημόσιων νεφούπολογιστικών υποδομών και υπηρεσιών νέφους, παρατηρήθηκε πως η πληθώρα των επιστημονικών συγγραμμάτων εξετάζει ανεξάρτητα ή σε αντιπαράβολη τις υποδομές φιλοξενίας και διαχείρισης εικονικών μηχανών (IaaS) από τις υποδομές φιλοξενίας και ενορχήστρωσης περιεκτών (PaaS - CaaS). Το γεγονός αυτό επιτρέπει, στον μελετητή, την πληρέστερη ανασκόπηση και εμβάθυνση στην κάθε τεχνολογία ξεχωριστά, αλλά δεν καλύπτει την από κοινού εξέταση τους και τα ζητήματα που προκύπτουν από την συνύπαρξη των δύο

τεχνολογιών στην ίδια υποδομή. Η οπτική της εφαρμογής των τεχνολογιών αυτών σε διαφορετικά επίπεδα (tiers), δηλαδή, η υλοποίηση της υποδομής ενορχήστρωσης περιεκτών πάνω από την υποδομή υπερεποπτιών εικονικών μηχανών, μπορεί να υπονοήσει ότι η συνύπαρξη των δύο τεχνολογιών καλύπτεται από την βιβλιογραφία. Στην προσπάθεια συνύπαρξης των δύο τεχνολογιών γίνεται εύκολα αντιληπτό πως αυτό μπορεί να έχει εφαρμογή σε μεγάλης κλίμακας υποδομές, ενώ προκαλεί πολλαπλά ζητήματα προς διαχείριση, όταν εφαρμόζεται σε μικρής κλίμακας υποδομές, με εντονότερα τα ζητήματα της επάρκειας και της αποτελεσματικής διαχείρισης των υπάρχοντων υπολογιστικών πόρων.

Ένας ακόμα περιορισμός που αφορά την μελέτη περίπτωσης και επηρεάζει το βάθος και την ευρύτητα της έρευνας, σχετίζεται με την διαθεσιμότητα ανθρώπινων και υλικών πόρων που αξιοποιήθηκαν κατά της διάρκεια της μελέτης αυτής. Ο φορέας, στον οποίο εκπονήθηκε η μελέτη, χαρακτηρίζεται ως μεσαίου μεγέθους οργανισμός, για τα ελληνικά δεδομένα ή μικρός για τα διεθνή δεδομένα. Αυτό έχει ως συνέπεια την αδυναμία παροχής πληθώρας ανθρώπινων και οικονομικών πόρων. Σε συνδυασμό με τις μικρής κλίμακας ιδιωτικές νεφούπολογιστικές υποδομές, που διαθέτει, οι υλοποιήσεις που μπορούν να εξεταστούν και οι δοκιμές που μπορούν να πραγματοποιηθούν, είναι περιορισμένες και εντός συγκεκριμένων ορίων. Αναλυτικότερα, η εφαρμογή των τεχνολογικών, αυτών, λύσεων απαιτεί σημαντική διάθεση ανθρώπινων και υλικών πόρων, ενώ δεν μπορεί να παραληφθεί η σημασία του χρονικού πλαισίου στο οποίο πρέπει να ολοκληρωθούν όλες οι εργασίες, της διαδικασίας μετασχηματισμού της υποδομής, όπως η μελέτη των διαθέσιμων λύσεων και η εγκατάσταση αυτών. Ο περιορισμός αυτός ενισχύεται αν ληφθεί υπόψιν, η επιβάρυνση που προκύπτει από την ταυτόχρονη παραγωγική λειτουργία των υποδομών του οργανισμού, που τις διαχειρίζεται το ίδιο ανθρώπινο δυναμικό.

Τέλος, τροχοπέδη στην έρευνα αυτή αποτέλεσε το περιορισμένο εύρος τεχνολογικών λύσεων, ανοιχτού κώδικα, που μπορούν να υποστηρίξουν την παράλληλη φιλοξενία εικονικών μηχανών και περιεκτών στην ίδια φυσική υποδομή και με δυνατότητες διαχείρισης και ενορχήστρωσης. Τα περισσότερα διαθέσιμα λογισμικά υποστηρίζουν κυρίως μεγάλες υποδομές, δεν υπάρχουν ώριμα λογισμικά για την παράλληλη φιλοξενία

Οι περισσότερες διαθέσιμες και ώριμες πλατφόρμες υποστηρίζουν ξεχωριστά την κάθε τεχνολογία, ενώ αυτές που τις υποστηρίζουν παράλληλα φαίνεται να είναι ακόμα πρώιμες ή να βασίζονται σε μη ώριμα λογισμικά. Επιπλέον, οι περισσότερες λύσεις προϋποθέτουν μεγάλης κλίμακας υποδομές, ενώ οι λύσεις που προορίζονται για μικρότερης κλίμακας υποδομές, στερούνται δυνατοτήτων. Παράλληλα, η τάση ενσωμάτωσης των υπηρεσιών σε νεφούπολογιστικές υποδομές, καθοδηγεί την ανάπτυξη των λογισμικών αυτών και συμπαρασύρει τα προαπαιτούμενα (requirements) και τις δυνατότητες των λογισμικών αυτών, που σχετίζονται άμεσα με την τάξη μεγέθους των υποδομών. Από την άλλη πλευρά, η καθιέρωση λειτουργιών και δυνατοτήτων που παρέχονται από τα επικρατέστερα λογισμικά σε συνδυασμό με τις απαιτήσεις των χρηστών για συνεχή ανάπτυξη και ενσωμάτωση (CI/CD) και για υψηλή διαθεσιμότητα των υπηρεσιών, μετακυλίου την ανάγκη υποστήριξης τους και από μικρότερης κλίμακας υποδομές.

5.3 Ερευνητικό Κενό και Προτάσεις για Μελλοντική Έρευνα

Η έρευνα στο πεδίο της εικονικοποίησης στο επίπεδο του υλισμικού (εικονικές μηχανές - IaaS) και στο επίπεδο του λειτουργικού συστήματος (περιέκτες – PaaS - SaaS) έχει ενταθεί σημαντικά τα τελευταία χρόνια. Αυτό προκύπτει, εξαιτίας της συνεχούς αυξανόμενης τάσης μετάβασης των υπηρεσιών σε νεφούπολογιστικές υποδομές, σε συνδυασμό με την εδραίωση των τεχνικών της συνεχούς ανάπτυξης παράδοσης (CI/CD) που αξιοποιούν τις τεχνολογίες των μικροϋπηρεσιών και των περιεκτών. Παρόλα αυτά, οι μελέτες φαίνεται να επικεντρώνονται σε μεγάλης κλίμακας υποδομές που υποστηρίζονται από μεγάλα και κατανεμημένα κέντρα δεδομένων ή από δημόσιους παρόχους νεφούπολογιστικών υποδομών. Σε αντιστοιχία με αυτό, οι πάροχοι λογισμικού κατευθύνουν την παραγωγή λογισμικού με κύριο στόχο την αποτελεσματική κάλυψη των αναγκών μεγάλης κλίμακας υποδομών. Η διαπίστωση αυτή επιβεβαιώνεται από την ανασκόπηση της διαθέσιμης επιστημονικής βιβλιογραφίας, που επικεντρώνεται στην μελέτη των τεχνολογιών αυτών, είτε ανεξάρτητα, είτε σε αντιπαραβολή μεταξύ τους (VMs vs Containers), υπό το πρίσμα χρήσης τους σε δημόσια νέφη και μεγάλες νεφούπολογιστικές υποδομές.

Παράλληλα με την αξιοποίηση των δημόσιων παρόχων υπηρεσιών νέφους και την φιλοξενία υπηρεσιών σε μεγάλης κλίμακας κέντρα δεδομένων, προκύπτει

ανάγκη φιλοξενίας υπηρεσιών νέφους και σε μικρότερης κλίμακας ιδιωτικές υποδομές. Οι μικρής κλίμακας ιδιωτικές υποδομές, συχνά διαθέτουν περιορισμένους πόρους για την ανάπτυξη των νεφοϋπολογιστικών συστημάτων τους, ενώ ταυτόχρονα καλούνται να συγκεράσουν την φιλοξενία εικονικών μηχανών (IaaS) και περιεκτών (PaaS – CaaS) εντός της ίδιας «περιορισμένης» φυσικής υποδομής, χωρίς να στερούνται τις συνήθεις δυνατότητες που προσφέρουν οι προαναφερθέντες πάροχοι. Το γεγονός αυτό δημιουργεί ένα ερευνητικό κενό αναφορικά με την μελέτη σχεδίασης, ανάπτυξης και υλοποίησης υποδομών παράλληλης διαχείρισης εικονικών μηχανών και ενορχήστρωσης περιεκτών, σε μικρής κλίμακας φυσικές υποδομές. Η μελέτη αυτή μπορεί να αφορά την χρήση κοινού λογισμικού και για τις δύο τεχνολογίες ή και την στρωματοποίηση (tiering) τους, με την χρήση διαφορετικού (π.χ. OKD over Proxmox) ή συμβατού (π.χ. Rancher over Harvester) λογισμικού. Επίσης, είναι σημαντικό να προσδιορίζεται η κλίμακα μεγέθους των υποδομών, της εκάστοτε μελέτης, ώστε να διασφαλίζεται η καταλληλότητα της τεχνολογικής λύσης συγκριτικά με το μέγεθος της υποδομής.

Η αναντιστοιχία φυσικής υποδομής και εφαρμοζόμενης τεχνολογικής λύσης εμπεριέχει σοβαρούς κινδύνους και ενδεχόμενα κόστη σε μικρομεσαίες επιχειρήσεις και οργανισμούς. Η μειωμένη απόδοση, η έλλειψη ελαστικότητας, η αδυναμία φιλοξενίας υπηρεσιών και η μη υποστήριξη υψηλής διαθεσιμότητας αποτελούν πιθανές συνέπειες της αναντιστοιχίας αυτής και μπορούν να προκαλέσουν σοβαρές αστοχίες στις προσφερόμενες υπηρεσίες.

Η παράλληλη φιλοξενία εικονικών μηχανών και περιεκτών, σε μικρής κλίμακας ιδιωτικές υποδομές, είναι ένα σύνθετο πρόβλημα που έχει λάβει ελάχιστη προσοχή τόσο στην ακαδημαϊκή όσο και στην επιχειρησιακή έρευνα. Η έλλειψη βιβλιογραφίας μπορεί να οφείλεται και στην έλλειψη κινήτρων για έρευνα προς αυτή την κατεύθυνση. Οι μικρές επιχειρήσεις και οι οργανισμοί συχνά δεν έχουν τους πόρους για να επενδύσουν σε έρευνα και ανάπτυξη, ενώ οι μεγάλοι οργανισμοί επενδύουν ευκολότερα σε τομείς που τους εξυπηρετούν.

Παρά τα εμπόδια αυτά, η έρευνα στην παράλληλη φιλοξενία εικονικών μηχανών και περιεκτών σε μικρής κλίμακας ιδιωτικές υποδομές είναι σημαντική για τις μικρές επιχειρήσεις. Η ανάπτυξη αποτελεσματικών λύσεων σε αυτό το πρόβλημα μπορεί να βοηθήσει τους οργανισμούς να βελτιώσουν την απόδοσή τους, να μειώσουν τα κόστη τους και να αυξήσουν την ασφάλειά τους.

Ως συνέπεια των περιορισμών της παρούσας μελέτης και του ερευνητικού κενού που καταγράφηκε, παρατίθενται προτάσεις για μελλοντική έρευνα, στοχευμένης σε υποδομές μικρής κλίμακας:

- Μελέτη των διαθέσιμων λογισμικών / πλατφορμών ανοιχτού κώδικα, για τη συνφιλοξενία εικονικών μηχανών και περιεκτών (υπό το πλαίσιο της ολοκληρωμένης διαχείρισης τους)
- Συγκριτική μελέτη της αρχιτεκτονικής των τεχνολογικών λύσεων συνφιλοξενίας εικονικών μηχανών και περιεκτών
- Μελέτη απόδοσης και αποτελεσματικής λειτουργίας των τεχνολογικών λύσεων συνφιλοξενίας εικονικών μηχανών και περιεκτών και σύγκριση τους με ανεξάρτητες / παραδοσιακές υποδομές εικονικών μηχανών και περιεκτών

5.4 Συμπεράσματα

Η παρούσα διπλωματική εργασία εστίασε στην μελέτη της επίδρασης της χρήσης των περιεκτών και της ανάγκης μετασχηματισμού των ιδιωτικών νεφοϋπολογιστικών υποδομών. Μέσα από μια ολοκληρωμένη βιβλιογραφική ανασκόπηση και την πραγματοποίηση μιας μελέτης περίπτωσης σε μια νεφοϋπολογιστική υποδομή ενός Ελληνικού Ερευνητικού Κέντρου, αναδείχθηκαν κάποια σημαντικά συμπεράσματα.

Στο πλαίσιο της μελέτης, εξετάστηκε η αναγκαιότητα της παράλληλης υποστήριξης εικονικών μηχανών και περιεκτών σε μία νεφοϋπολογιστική υποδομή. Από τα αποτελέσματα προέκυψε ότι η παράλληλη χρήση και υποστήριξη των δύο τεχνολογιών αποτελεί ένα βασικό στοιχείο για τη δημιουργία περισσότερο σύγχρονων, ευέλικτων και αποτελεσματικών υποδομών, προσφέροντας τη δυνατότητα απομόνωσης και ευκολότερης μεταφοράς των εφαρμογών και υποστήριξης των σύγχρονων μεθοδολογιών ανάπτυξης λογισμικού (Agile, Scrum CI/CD κλπ.). Ωστόσο, ένα ερευνητικό κενό που προέκυψε από τη μελέτη, αφορά την έλλειψη επαρκούς βιβλιογραφίας που αναφέρεται στην παράλληλη χρήση εικονικών μηχανών και περιεκτών σε μικρής κλίμακας υποδομές, κάτι που αναδεικνύει τη σημασία των μελλοντικών ερευνών που θα ασχοληθούν με αυτό το θέμα.

Η ανασκόπηση της διαθέσιμης ποικιλίας λογισμικού και πλατφορμών αποκάλυψε την πληθώρα των επιλογών που υπάρχουν για την υποστήριξη διαφορετικών και ανεξάρτητων υποδομών (IaaS – PaaS - CaaS). Ωστόσο, η εύρεση του κατάλληλου λογισμικού, που θα προσφέρει την καλύτερη λύση για την εκάστοτε περίπτωση, είναι μια πρόκληση και απαιτεί προσεκτική εξέταση και αξιολόγηση. Επομένως, συνιστάται η συνεργασία με εξειδικευμένους εμπειρογνώμονες για την επιλογή της κατάλληλης πλατφόρμας, που θα ικανοποιεί τις ανάγκες και τις απαιτήσεις της συγκεκριμένης υποδομής.

Ενώ οι διαθέσιμες πλατφόρμες φιλοξενίας ανεξάρτητων υποδομών διαχείρισης εικονικών μηχανών ή ενορχήστρωσης περιεκτών, μπορούν να χαρακτηριστούν εδραιωμένες, σταθερές και αξιόπιστες, η υποστήριξη φιλοξενίας εικονικών μηχανών σε υποδομές ενορχήστρωσης περιεκτών απαιτεί ωρίμανση και βελτιστοποίηση των ήδη υπάρχοντων λογισμικών, ώστε να αξιοποιηθούν παραγωγικά προσφέροντας σταθερότητα και ασφάλεια. Στο σημείο αυτό αξίζει να επισημανθεί ότι παρατηρείται αυξημένη δραστηριότητα στην παραγωγή και εξέλιξη τέτοιων λογισμικών, ενώ χαρακτηριστικό παράδειγμα αποτελεί η παρουσίαση του νέου λογισμικού TrueNAS Scale (Απρίλιος 2023), που επικαλείται την ενσωμάτωση της πληθώρας των απαιτούμενων λειτουργιών. Το συγκεκριμένο λογισμικό διατέθηκε σε σταθερή έκδοση κατά τη διάρκεια της συγγραφής της παρούσας εργασίας και δεν επετεύχθη η δοκιμή του.

Η μελέτη περίπτωσης κατέδειξε, πως ο μετασχηματισμός της νεφούπολογιστικής υποδομής, στο Ελληνικό Κέντρο Θαλασσιών Ερευνών, ολοκληρώθηκε επιτυχώς, επιβεβαιώνοντας τη δυνατότητα συνύπαρξης πλατφόρμας διαχείρισης υπερεποπών εικονικών μηχανών και ενορχηστρωτών περιεκτών στην ίδια υποδομή και υπό συνθήκες περιορισμένης διαθεσιμότητας πόρων. Παρόλα αυτά, οι δυσκολίες που προέκυψαν λόγω των περιορισμένων φυσικών και ανθρώπινων πόρων, καθώς και της περίπλοκης τεχνολογικής και λειτουργικής διαδικασίας, αντιμετωπίστηκαν με τη σχολαστική επιλογή των τεχνολογικών λύσεων και της κατάλληλης αρχιτεκτονικής επιπέδων (tiering – PaaS over IaaS), ώστε να διασφαλίζεται η ομαλή και σταθερή λειτουργία της υποδομής. Παράλληλα, προκύπτει η ανάγκη για περαιτέρω ενίσχυση των πόρων και για επιμόρφωση του προσωπικού στις τεχνολογίες αυτές, όπως και η διαρκής παρακολούθηση της εξέλιξης των τεχνολογιών, ώστε να επιλέγονται και να εφαρμόζονται οι βέλτιστες

και οι καταλληλότερες λύσεις, προκειμένου ο μετασχηματισμός των υποδομών να προσφέρει τη μέγιστη απόδοση, ευελιξία και αξιοπιστία.

Συμπερασματικά, η παρούσα διπλωματική εργασία ανέδειξε την σημασία της υποστήριξης περιεκτών και εικονικών μηχανών, υπό τη σκέπη μιας κοινής υποδομής, και την ανάγκη μετασχηματισμού των ιδιωτικών νεφοϋπολογιστικών υποδομών, προς την κατεύθυνση αυτή, υπογραμμίζοντας τα οφέλη και τις προκλήσεις που προκύπτουν. Μέσω περαιτέρω έρευνας και βελτιστοποίησης, οι πλατφόρμες συνφιλοξενίας περιεκτών και εικονικών μηχανών δύναται να αποτελέσουν το βασικό εργαλείο παροχής υπηρεσιών νέφους, ακόμα και για μικρομεσαίους οργανισμούς και επιχειρήσεις. Η αξιοποίηση των υποδομών αυτών, εξασφαλίζει τις δυνατότητες και κυρίως την ιδιωτικότητα των νεφοϋπολογιστικών υποδομών, προωθεί την ανάπτυξη και την χρήση εφαρμογών τεχνολογίας νέφους, ενώ ταυτόχρονα διασφαλίζει την δυνατότητα μετάβασης σε οποιοδήποτε εναλλακτικό μοντέλο ανάπτυξης υπολογιστικής νέφους.

BIBΛΙΟΓΡΑΦΙΑ

- [1] Peter Mell και Timothy Grance, «The NIST Definition of Cloud,» *NIST Special Publication 800-145*, 2011.
- [2] F. Borko, «Cloud Computing Fundamentals,» σε *Handbook of Cloud Computing*, New York Dordrecht Heidelberg London, Springer, 2010.
- [3] X. Cheng, «Cloud Computing and Decision-Making Determinants, Modelling and Impacts,» *Universite PARIS-SACLAY*, pp. 14-25, 2017.
- [4] W. Jill, «Introduction to Cloud Computing,» σε *CompTIA Cloud+ Guide to Cloud Computing*, Boston, Cengage Learning, Inc, 2021, p. 14.
- [5] D. Freet, R. Agrawal, J. Sherin και J. J. Walker , «Cloud Forensics Challenges from a Service Model Standpoint: IaaS, PaaS and SaaS,» *MEDES '15: Proceedings of the 7th International Conference on Management of computational and collective intelligence in Digital EcoSystems*, p. Pages 148–155, 2015.
- [6] W. Chai, «Software as a Service (SaaS),» TechTarget, 2022. [Ηλεκτρονικό]. Available:
<https://www.techtarget.com/searchcloudcomputing/definition/Software-as-a-Service>.
- [7] Anthony T. Velte, Toby Velte και R. Elsenpeter, *Cloud Computing: A Practical Approach*, McGraw-Hill, 2010.
- [8] A. Singh, *Cloud Computing Simply in Depth*, Kindle Direct Publishing - Amazon, 2021-22.
- [9] M. Gilje, J. G. Zhao και C. Rong, *Cloud Computing*, Beijing, China: Springer, 2009.
- [10] A. Lisdorf, *Cloud Computing Basics: A Non-Technical Introduction*, Copenhagen, Denmark: Apress, 2021.

- [11] D. C. Marinescu, *Cloud Computing Theory and Practice*, Cambridge, United States: Morgan Kaufmann - Katey Birtcher, 2018.
- [12] A. Paul, M. K. Anvekar, R. Jacob και K. C. Sekaran, «Cyber Forensics in Cloud Computing,» *The International Institute for Science, Technology and Education (IISTE)*, 2012.
- [13] E. Simmon, *Evaluation of Cloud Computing Services Based on NIST*, NIST Cloud Computing Cloud Services Working Group, 2018.
- [14] «Cloud Computing vs Virtualization: Definitions, Pros and Cons, and Key Differences,» apriorit, 2021. [Ηλεκτρονικό]. Available: <https://www.apriorit.com/white-papers/373-cloud-computing-vs-virtualization>.
- [15] M. G. Jaatun, *Cloud Computing*, Beijing, China: Springer, 2009.
- [16] D. Zomaya, «Container vs. Hypervisor: What's the Difference?,» CBTNuggets, 21 October 2019. [Ηλεκτρονικό]. Available: <https://www.cbtnuggets.com/blog/certifications/cloud/container-v-hypervisor-whats-the-difference>.
- [17] «ΣΥΣΤΗΜΑ ΑΝΑΠΤΥΞΗΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ ΕΙΚΟΝΟΠΟΙΗΣΗΣ ΒΑΣΙΣΜΕΝΗΣ ΣΕ ΠΕΡΙΕΚΤΕΣ ΜΕ ΒΑΣΗ ΤΟ ΛΟΓΙΣΜΙΚΟ KUBERNETES,» ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ, 2019. [Ηλεκτρονικό]. Available: <http://ikee.lib.auth.gr/record/306394/files/GRI-2019-25145.pdf>.
- [18] Y. Zhang, *Future Wireless Networks and Information Systems*, Wuhan, China: Springer, 2012.
- [19] M. Kaschke, «Virtual Machine (VM) vs. Container,» Medium, 24 Nov 2020. [Ηλεκτρονικό]. Available: <https://mkaschke.medium.com/virtual-machine-vm-vs-container-13ab51f4c177>.

- [20] M. R. Pamidi, «Impact of Containers on Data Center Virtualization,» *IMEX Research*, αρ. IT Newswire.
- [21] S. M. Jain, *Linux Containers and Virtualization A Kernel Perspective*, Bengaluru, India: Apress, 2020.
- [22] vmware, «What is a hypervisor?,» vmware, [Ηλεκτρονικό]. Available: <https://www.vmware.com/topics/glossary/content/hypervisor.html>.
- [23] J. Maning, «Hypervisor Type 1 vs. Type 2: What Is the Difference, and Does It Matter?,» www.makeuseof.com, 9 Jan 2023. [Ηλεκτρονικό]. Available: <https://www.makeuseof.com/type-1-vs-type-2-hypervisor-what-is-the-difference/>.
- [24] ResellerClub, «Type 1 and Type 2 Hypervisors: What Makes Them Different,» *ResellerClub - Medium*, 24 May 2019. [Ηλεκτρονικό]. Available: <https://medium.com/teamresellerclub/type-1-and-type-2-hypervisors-what-makes-them-different-6a1755d6ae2c>.
- [25] RedHat, «What is KVM?,» RedHat, 11 May 2022. [Ηλεκτρονικό]. Available: <https://www.redhat.com/en/topics/virtualization/what-is-KVM>.
- [26] Y. Pan, I. Chen, F. Brasileiro, G. Jayaputera και R. O. Sinnott, «A Performance Comparison of Cloud-based Container Orchestration Tools,» *IEEE International Conference on Big Knowledge*, 2019.
- [27] D. Bernstein, «Containers and Cloud: From LXC to Docker to Kubernetes,» *IEEE Cloud Computing*, τόμ. 1, αρ. 3, 2014.
- [28] P. Jain, «Containers for Beginners: What, Why and Types,» K21Academy, 29 April 2021. [Ηλεκτρονικό]. Available: <https://k21academy.com/docker-kubernetes/what-are-containers/#3>.
- [29] M. Kirov, «Digging Into Runtimes – runc,» Quarkslab's blog, 10 May 2022. [Ηλεκτρονικό]. Available: <https://blog.quarkslab.com/digging-into-runtimes-runc.html>.

- [30] I. Velichko, «What Is a Standard Container: Diving Into the OCI Runtime Spec,» iximiuz, 18 12 2022. [Ηλεκτρονικό]. Available: <https://iximiuz.com/en/posts/oci-containers/>.
- [31] P. Jain, «Containers for Beginners: What, Why and Types,» K21Academy, 29 April 2021. [Ηλεκτρονικό]. Available: <https://k21academy.com/docker-kubernetes/what-are-containers/#3>.
- [32] M. Moravcik, P. Segec, M. Kontsek, J. Uramova και J. Papan, «Comparison of LXC and Docker technologies,» *18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2020.
- [33] C. Miller, «Linux Containers on the Raspberry Pi,» makikiweb, 3 Jan 2019. [Ηλεκτρονικό]. Available: http://www.makikiweb.com/Pi/lxc_on_the_pi.html.
- [34] T. Banerjee, «Understanding the key differences between LXC and Docker,» flockport, 9 Aug 2014. [Ηλεκτρονικό]. Available: <https://archives.flockport.com/lxc-vs-docker/>.
- [35] Preeth E N, Fr. Jaison Paul Mulericka, Biju Paul και Yedhu Sastri, «Evaluation of Docker containers based on hardware utilization,» *International Conference on Control Communication & Computing India (ICCC)*, 2015.
- [36] c. |. DockerLab, «Beginners Track - What happens when Containers are Launched?,» collabnix | DockerLab, [Ηλεκτρονικό]. Available: <https://dockerlabs.collabnix.com/beginners/container-runtime.html>.
- [37] E. Baker, «Kubernetes containers: A comprehensive runtime comparison,» capitalone, 10 Jun 2020. [Ηλεκτρονικό]. Available: <https://www.capitalone.com/tech/cloud/container-runtime/>.
- [38] I. M. A. Jawarneh, P. Bellavista, F. Bosi και L. Foschini, «Container Orchestration Engines:A Thorough Functional and Performance Comparison».

- [39] L. Mercl και J. Pavlik, «The Comparison of Container Orchestrators,» Hradec Kralove, Czech Republic, 2018.
- [40] C. Malmberg, «Container Orchestration Tools Comparison,» RAFAY, 02 June 2022. [Ηλεκτρονικό]. Available: <https://rafay.co/the-kubernetes-current/author/cliff/>.
- [41] A. Allen, «Definition Apache Mesos,» TechTarget, July 2016. [Ηλεκτρονικό]. Available: <https://www.techtarget.com/searchitoperations/definition/Apache-Mesos>.
- [42] «Qemu Driver,» Hashicorp, [Ηλεκτρονικό]. Available: <https://developer.hashicorp.com/nomad/docs/drivers/qemu>.
- [43] «Introduction to Nomad,» Hashicorp, [Ηλεκτρονικό]. Available: <https://developer.hashicorp.com/nomad/intro>.
- [44] «What are Containers?,» Google, [Ηλεκτρονικό]. Available: <https://cloud.google.com/learn/what-are-containers>.
- [45] A. Khan, «Key Characteristics of a Container Orchestration Platform to Enable Modern Application,» *IEEE Cloud Computing*, 2017.
- [46] A. Puliafito και K. S. Trivedi, *Systems Modeling*, Springer, 2019.
- [47] Eurostat, «Cloud computing used by 42% of enterprises,» Eurostat, 2021. [Ηλεκτρονικό]. Available: <https://ec.europa.eu/eurostat/web/products-eurostat-news/-/ddn-20211209-2>.
- [48] Eurostat, «Cloud computing - statistics on the use by enterprises,» 2021. [Ηλεκτρονικό]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Cloud_computing_-_statistics_on_the_use_by_enterprises#Use_of_cloud_computing_highlights.

- [49] D. Linthicum, «Practical use of microservices in moving workloads to the cloud,,» *IEEE Cloud Computing*, τόμ. 3, αρ. 5, 2016.
- [50] Mohamed K. Hussein, Mohamed H. Mousa και Mohamed A. Alqarni , «A placement architecture for a container as a service (CaaS) in a cloud environment,» *Journal of Cloud Computing*, τόμ. 8, pp. 1-15, 2019.
- [51] Y. S. D. Reddy, P. S. Reddy, N. Ganesan και B. Thangaraju, «Performance Study of Kubernetes Cluster Deployed on Openstack, VMs and BareMetal,» σε *IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, India, 2022.
- [52] C. Tozzi, «Kubernetes on Bare Metal: Why and How,» 2021. [Ηλεκτρονικό]. Available: <https://platform9.com/blog/kubernetes-on-bare-metal-why-and-how>.
- [53] D. Olaogun, «Making the Right Choice: Kubernetes on Virtual Machines vs. Bare Metal,» EQUINIX, 2023. [Ηλεκτρονικό]. Available: <https://deploy.equinix.com/blog/making-the-right-choice-kubernetes-on-virtual-machines-vs-bare-metal/>.
- [54] Lingayat, Ashish, Ranjana R. Badre και Anil Kumar, «Integration of linux containers in openstack: An introspection.,» *Indonesian Journal of Electrical Engineering and Computer Science*, τόμ. 12, αρ. 3, pp. 1094-1105, 2018.
- [55] K. Nelson, «How to use Kubernetes and OpenStack together,» opensource.com, 2022. [Ηλεκτρονικό]. Available: <https://opensource.com/article/22/3/kubernetes-openstack>.
- [56] S. F. Piraghaj, A. V. Dastjerdi, R. N. Calheiros και R. Buyya, «Efficient Virtual Machine Sizing for Hosting Containers as a Service,» σε *IEEE World Congress on Services*, New York, USA, 2015.
- [57] Muhammad Alvin και Rizal Fathoni Aji, «DevOps Implementation with Enterprise On-Premise Infrastructure,» *JURNAL MEDIA INFORMATIKA BUDIDARMA*, τόμ. 7, αρ. 1, pp. 331-341, 2023.

- [58] Junzo Watada, Arunava Roy, Raturaj Uday Kadikar και Hoang Pham, «Emerging Trends, Techniques and Open Issues of Containerization: A Review,» *IEEE Access*, τόμ. 7, pp. 152443-152472, 2019.
- [59] Gartner, «Gartner Identifies Five Ways to Migrate Applications to the Cloud,» 2011. [Ηλεκτρονικό]. Available: <https://www.pressebox.com/pressrelease/gartner-uk-ltd/Gartner-Identifies-Five-Ways-to-Migrate-Applications-to-the-Cloud/boxid/424552>.
- [60] S. Orban, «6 Strategies for Migrating Applications to the Cloud,» AWS, 2016. [Ηλεκτρονικό]. Available: <https://aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/>.
- [61] A. L. Shastry, B. Prathima, C. P. Ramya και P. Hallymysore, «Approaches for migrating non cloud-native applications to the cloud,» *EEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0632-0638, 2022.
- [62] B. Rudisail, «How to Migrate Virtual Machines to Containers: A Five-step Guide,» Spiceworks, 2022. [Ηλεκτρονικό]. Available: <https://www.spiceworks.com/tech/devops/articles/steps-to-migrate-from-vm-to-container/>.
- [63] Yin και Robert K., *Case Study Research: desing and methods*, Sage Publications, 2009.
- [64] H.-G. Ridder, «The theory contribution of case study research designs,» *Springer*, 2017.
- [65] K. F. Pearlson, C. S. Saunders και D. S. Galletta, *Managing and Using Information Systems: A Strategic Approach*, WILEY, 2020.
- [66] E. Turban, C. Pollard και G. Wood, *Information Technology for Management: On-Demand Strategies for Performance, Growth, and Sustainability*, WILEY, 2018.

- [67] R. s. Shandu, «Role-based Access Control,» σε *ADVANCES IN COMPUTERS*, ELSEVIER, 1998, pp. 237-286.
- [68] «What is KubeSphere,» 28 08 2023. [Ηλεκτρονικό]. Available: <https://www.kubesphere.io/docs/v3.3/introduction/what-is-kubesphere/>.
- [69] «Support Matrix for OpenShift Container Platform IPI Installation on Red Hat Virtualization,» Red Hat, 18 July 2022. [Ηλεκτρονικό]. Available: <https://access.redhat.com/articles/5485861>.
- [70] «Example: Deploying WordPress and MySQL with Persistent Volumes,» 24 August 2023. [Ηλεκτρονικό]. Available: <https://kubernetes.io/docs/tutorials/stateful-application/mysql-wordpress-persistent-volume/>.
- [71] «The Raft Consensus Algorithm,» [Ηλεκτρονικό]. Available: <https://raft.github.io/>.
- [72] «Restarting the cluster gracefully,» [Ηλεκτρονικό]. Available: https://docs.okd.io/latest/backup_and_restore/graceful-cluster-restart.html.