



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

**Εγκατάσταση, λειτουργία και παρακολούθηση περιβάλλοντος
Homelab με τεχνολογίες Proxmox, Docker και Kubernetes**

Πτυχιακή Εργασία

του

Ματσούκα Δημήτριου Παρασκευά
Α.Μ. 1507

Επιβλέπων : Γκόγκος Χρήστος

Άρτα, Ιούνιος 2025

Πρόλογος και Ευχαριστίες

Θα ήθελα να εκφράσω την εγκάρδια ευγνωμοσύνη μου στον επιβλέποντα καθηγητή, κ. Γκόγκο Χρήστο, για την πολύτιμη καθοδήγησή του καθ' όλη τη διάρκεια συγγραφής της πτυχιακής αυτής εργασίας. Οι γνώσεις, η εμπειρία και οι εποικοδομητικές του παρατηρήσεις συνέβαλαν στην ποιοτική αναβάθμισή της, ενώ η αμέριστη υποστήριξη και η υπομονή του με ενθάρρυναν να ξεπεράσω κάθε πρόκληση. Του οφείλω θερμές ευχαριστίες για το χρόνο και το ενδιαφέρον που αφιέρωσε στην επιτυχία αυτού του εγχειρήματος.

Πίνακας Περιεχομένων

1 Εισαγωγή.....	12
1.1 Αντικείμενο Πτυχιακής	12
1.2 Δομή Πτυχιακής.....	12
1.3 Ο όρος Homelab.....	12
2 Τεχνολογίες Εικονικοποίησης, Κοντέινερ και Ενορχήστρωσης.....	13
2.1 Τεχνολογίες Εικονικοποίησης (Virtualization)	13
2.2 Βασικές Αρχές των Κοντέινερ (Containerization).....	14
2.3 Kubernetes και Ενορχήστρωση Κοντέινερ (Orchestration).....	14
3 Σχεδίαση και Αρχιτεκτονική.....	16
3.1 Απαιτήσεις Συστήματος και Περιορισμοί.....	16
3.2 Διάταξη Υλικού και Δικτύων	17
3.3 Επιλογή Λειτουργικού συστήματος και Λογισμικού	17
3.3.1 Proxmox Virtual Environment (PVE)	17
3.3.2 Docker & Docker Compose	17
3.3.3 K3s Kubernetes	18
3.3.4 Gatus.....	19
3.4 Ζητήματα Ασφαλείας.....	19
4 Υλοποίηση.....	20
4.1 Εγκατάσταση και Ρύθμιση του Proxmox VE	20
4.1.1 Λήψη του είδωλου εγκατάστασης (ISO).....	20
4.1.2 Εγγραφή του είδωλου εγκατάστασης (ISO) σε μονάδα USB	21
4.1.3 Εγκατάσταση του Proxmox.....	22
4.1.4 Πρόσβαση στο διαχειριστικό περιβάλλον του Proxmox.....	30
4.1.5 Απαραίτητες ρυθμίσεις μετά την εγκατάσταση του Proxmox	32
4.1.6 Αναβάθμιση του Proxmox στην τελευταία έκδοση.....	34
4.2 Δημιουργία και Ρύθμιση των Εικονικών Μηχανών και Κοντέινερ	36
4.2.1 Διαδικασία εισαγωγής ειδώλου εγκατάστασης (ISO) στο Proxmox.....	36
4.2.2 Διαδικασία εισαγωγής προτύπου LXC κοντέινερ στο Proxmox.....	37
4.2.3 Διαδικασία δημιουργίας εικονικής μηχανής στο Proxmox	38
4.2.4 Διαδικασία δημιουργίας LXC κοντέινερ στο Proxmox.....	44
4.2.5 Διαχείριση εικονικών μηχανών και κοντέινερ μέσα από το Proxmox	51
4.3 Εγκατάσταση και Ρύθμιση του Docker.....	53
4.4 Ανάπτυξη Kubernetes Cluster	54
4.5 Υλοποίηση Συστήματος Παρακολούθησης (Monitoring).....	54
5 Αντίγραφα Ασφαλείας και Αξιολόγηση Απόδοσης.....	59
5.1 Αντίγραφα Ασφαλείας	59
5.1.1 Δημιουργία αντίγραφου ασφάλειας στο Proxmox	59
5.1.2 Επαναφορά αντίγραφου ασφάλειας στο Proxmox	61
5.2 Παρακολούθηση και Ανάλυση Χρήσης Πόρων.....	62
5.2.1 Σύγκριση χρήσης πόρων LXC κοντέινερ και εικονικής μηχανής	62

6 Παρουσίαση Προσωπικού Homelab.....	64
6.1 Διάγραμμα Υποδομής.....	64
6.2 Γενική Περιγραφή	64
6.2.1 Υπολογιστική Υποδομή.....	65
6.2.2 Δικτυακή Υποδομή	66
6.2.3 Υπηρεσίες που φιλοξενώ	67
6.2.4 Φωτογραφίες Υποδομής.....	69
6.3 Χρήσιμες Διαδικασίες	70
6.3.1 SSH (Secure Shell)	70
6.3.2 Proxmox GPU Passthrough.....	71
7 Συμπεράσματα	73
8 Βιβλιογραφία – Πηγές.....	74
9 Παραρτήματα.....	75
9.1 Παράρτημα 1.....	75

Πίνακας Εικόνων

Εικόνα 1 Διάφορες μεταξύ της παραδοσιακής αρχιτεκτονικής και της εικονικής αρχιτεκτονικής.	13
Εικόνα 2 Διάφορες μεταξύ της εικονικής αρχιτεκτονικής και της αρχιτεκτονικής κοντεϊνερ.	14
Εικόνα 3 Παράδειγμα ενός Dockerfile το οποίο δημιουργεί έναν διακομιστή ιστού nginx....	18
Εικόνα 4 Λήψη του ειδώλου εγκατάστασης (ISO) από την επίσημη ιστοσελίδα του Proxmox.	20
Εικόνα 5 Διαδικασία εύρεσης της μονάδας USB και εγγραφή του ειδώλου εγκατάστασης (ISO) σε αυτή στα Linux.	22
Εικόνα 6 Οδηγός εγκατάστασης του Proxmox.	23
Εικόνα 7 Συμφωνία αδειοδότησης τελικού-χρήστη Proxmox.	24
Εικόνα 8 Επιλογή σκληρού δίσκου κατά την εγκατάσταση του Proxmox.	25
Εικόνα 9 Επιλογή ρυθμίσεων τοποθεσίας και ζώνης ώρας κατά την εγκατάσταση του Proxmox.	26
Εικόνα 10 Εισαγωγή διαπιστευτηρίων και διεύθυνση ηλεκτρονικού ταχυδρομείου κατά την εγκατάσταση του Proxmox.	27
Εικόνα 11 Επιλογή ρυθμίσεων δικτύου κατά την εγκατάσταση του Proxmox.	28
Εικόνα 12 Σύνοψη των επιλογών μας πριν την τελική εγκατάσταση του Proxmox.	29
Εικόνα 13 Επιτυχής εγκατάσταση του Proxmox.	30
Εικόνα 14 Κονσόλα ελέγχου του Proxmox.	30
Εικόνα 15 Σφάλμα πιστοποιητικών κατά την πρόσβαση στο διαχειριστικό περιβάλλον του Proxmox.	31
Εικόνα 16 Είσοδος διαπιστευτηρίων κατά την πρόσβαση στο διαχειριστικό περιβάλλον του Proxmox.	31
Εικόνα 17 Προειδοποίηση για έλλειψη αδειας κατά την πρόσβαση στο διαχειριστικό περιβάλλον του Proxmox.	32
Εικόνα 18 Κεντρική σελίδα διαχειριστικού περιβάλλοντος του Proxmox.	32
Εικόνα 19 Διαχείριση των ενημερώσεων του Proxmox.	33
Εικόνα 20 Απενεργοποίηση των enterprise APT repositories στο Proxmox.	33
Εικόνα 21 Ενεργοποίηση του No-Subscription APT repository στο Proxmox.	34
Εικόνα 22 Επιτυχής ενεργοποίηση των ενημερώσεων του Proxmox χωρίς ενεργή συνδρομή.	34
Εικόνα 23 Πρώτο βήμα ενημέρωσης του Proxmox στην τελευταία έκδοση.	35
Εικόνα 24 Δεύτερο βήμα ενημέρωσης του Proxmox στην τελευταία έκδοση.	35
Εικόνα 25 Επιτυχής ενημέρωση του Proxmox στην τελευταία έκδοση.	36
Εικόνα 26 Διαδικασία εισαγωγής ενός ειδώλου εγκατάστασης (ISO) στο Proxmox.	36
Εικόνα 27 Λήψη του ειδώλου εγκατάστασης (ISO) στο Proxmox.	37
Εικόνα 28 Διαδικασία εισαγωγής ενός προτύπου LXC κοντεϊνερ στο Proxmox.	37
Εικόνα 29 Λήψη ενός προτύπου LXC κοντεϊνερ μέσα από την λίστα του Proxmox.	38
Εικόνα 30 Κουμπί δημιουργίας εικονικής μηχανής στο διαχειριστικό περιβάλλον του Proxmox.	38
Εικόνα 31 Πρώτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.	39
Εικόνα 32 Δεύτερο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.	39
Εικόνα 33 Τρίτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.	40

Εικόνα 34 Τέταρτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.....	40
Εικόνα 35 Πέμπτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.....	41
Εικόνα 36 Έκτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.....	42
Εικόνα 37 Έβδομο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.....	42
Εικόνα 38 Όγδοο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.....	43
Εικόνα 39 Εγκατάσταση του λειτουργικού συστήματος Ubuntu 24.04.2 LTS Server στην εικονική μηχανή.....	43
Εικόνα 40 Κουμπί δημιουργίας LXC κοντέινερ στο διαχειριστικό περιβάλλον του Proxmox.....	44
Εικόνα 41 Πρώτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.....	44
Εικόνα 42 Δεύτερο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.....	45
Εικόνα 43 Τρίτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.....	46
Εικόνα 44 Τέταρτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.....	47
Εικόνα 45 Πέμπτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.....	48
Εικόνα 46 Έκτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.....	49
Εικόνα 47 Έβδομο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.....	50
Εικόνα 48 Όγδοο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.....	51
Εικόνα 49 Μενού διαχείρισης λειτουργιών εικονικών μηχανών και LXC κοντέινερ του διαχειριστικού περιβάλλοντος του Proxmox.....	52
Εικόνα 50 Λειτουργία κονσόλας του διαχειριστικού περιβάλλοντος του Proxmox.....	52
Εικόνα 51 Διαχείριση ενός λειτουργικού συστήματος χρησιμοποιώντας την λειτουργία του διαχειριστικού περιβάλλοντος του Proxmox.....	53
Εικόνα 52 Επιβεβαίωση της ορθής εγκατάστασης του Docker.....	53
Εικόνα 53 Επιβεβαίωση της ορθής εγκατάστασης του K3s cluster.....	54
Εικόνα 54 Εκτέλεση εντολών προετοιμασίας εγκατάστασης του Gatus Docker κοντέινερ.....	56
Εικόνα 55 Εκτέλεση εντολών για την εκκίνηση του Gatus κοντέινερ.....	57
Εικόνα 56 Η σελίδα του Gatus που υλοποιήσαμε.....	58
Εικόνα 57 Διαδικασία δημιουργίας αντίγραφου ασφαλείας μέσα από το διαχειριστικό περιβάλλον του Proxmox.....	59
Εικόνα 58 Δημιουργία αντίγραφου ασφαλείας με την μέθοδο «Snapshot» μέσα από το διαχειριστικό περιβάλλον του Proxmox.....	60
Εικόνα 59 Ολοκλήρωση διαδικασίας λήψης αντίγραφου ασφαλείας μέσα από το διαχειριστικό περιβάλλον του Proxmox.....	61
Εικόνα 60 Διαδικασία επαναφοράς αντίγραφου ασφαλείας μέσα από το διαχειριστικό περιβάλλον του Proxmox.....	61
Εικόνα 61 Επαναφορά αντίγραφου ασφαλείας μέσα από το διαχειριστικό περιβάλλον του Proxmox.....	62

Εικόνα 62 Κατανάλωση πόρων συστήματος του LXC κοντέινερ που υλοποιήθηκε στο υποκεφάλαιο 4.2.4.	63
Εικόνα 63 Κατανάλωση πόρων συστήματος της εικονικής μηχανής που υλοποιήθηκε στο υποκεφάλαιο 4.2.3.	63
Εικόνα 64 Σχεδιάγραμμα του προσωπικού μου homelab.....	64
Εικόνα 65 Στιγμιότυπο από την υπηρεσία Home Assistant που φιλοξενώ στο προσωπικό μου homelab.....	67
Εικόνα 66 Στιγμιότυπο από την υπηρεσία Nextcloud που φιλοξενώ στο προσωπικό μου homelab.....	68
Εικόνα 67 Στιγμιότυπο από την υπηρεσία Ollama που φιλοξενώ στο προσωπικό μου homelab.....	68
Εικόνα 68 Ο εξοπλισμός του homelab μου μέσα στο Rack.....	69
Εικόνα 69 Ο κόμβος 1 του homelab μου.	70

Λεξικό Όρων

AI	Artificial Intelligence
BIOS	Basic Input/Output System
CPU	Central Processing Unit
GB	Gigabyte
GiB	Gibibyte
GPU	Graphics Processing Unit
IOMMU	Input-Output Memory Management Unit
IoT	Internet of Things
ISO	Identical Storage image of Optical media
KVM	Kernel-based Virtual Machines
LLM	Large Language Model
LXC	Linux Containers
MB	Megabyte
MiB	Mebibyte
NAS	Network Attached Storage
NVME	Non-Volatile Memory Express
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
SBC	Single Board Computers
SR-IOV	Single Root IO Virtualization
SSD	Solid-State Drive
SSL	Secure Sockets Layer
TB	Terabyte
UEFI	Unified Extensible Firmware Interface
UPS	Uninterruptible Power Supply
VE	Virtual Environment
VFIO	Virtual Function I/O
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network

Περίληψη

Η παρούσα πτυχιακή εργασία ασχολείται με τη εγκατάσταση, λειτουργία και παρακολούθηση ενός περιβάλλοντος homelab (οικιακού εργαστηρίου) αξιοποιώντας σύγχρονες τεχνολογίες εικονικοποίησης (virtualization) και ενορχήστρωσης κοντέινερ (container orchestration). Συγκεκριμένα, χρησιμοποιήθηκαν οι πλατφόρμες Proxmox VE για εικονικοποίηση, Docker για την ανάπτυξη κοντέινερ και Kubernetes για την ενορχήστρωσή τους. Στόχος ήταν η υλοποίηση ενός ολοκληρωμένου περιβάλλοντος που να επιτρέπει την εύκολη δοκιμή νέων τεχνολογιών και την απόκτηση εμπειρίας στη διαχείριση σύγχρονων υποδομών. Στο πλαίσιο της εργασίας καταγράφονται οι απαιτήσεις του συστήματος, περιγράφονται βήμα προς βήμα οι διαδικασίες εγκατάστασης και παραμετροποίησης, καθώς και οι μέθοδοι παρακολούθησης της υποδομής. Τέλος, ακολουθεί μια παρουσίαση του προσωπικού μου παραγωγικού homelab η οποία περιέχει πληροφορίες για την φυσική υποδομή, την δικτυακή υποδομή, και κάποιες από τις υπηρεσίες που φιλοξενώ.

1

Εισαγωγή

1.1 Αντικείμενο Πτυχιακής

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός, η υλοποίηση και η διαχείριση ενός οικιακού εργαστηρίου (homelab) βασισμένου σε Proxmox, Docker και Kubernetes. Σκοπός είναι η εγκατάσταση ενός ευέλικτου και απομονωμένου περιβάλλοντος δοκιμών, στο οποίο ο χρήστης μπορεί να πειραματίζεται με νέες τεχνολογίες, να αξιολογεί και να δοκιμάζει υπηρεσίες και εφαρμογές, χωρίς τους περιορισμούς και το κόστος των cloud λύσεων. Παράλληλα, η υλοποίηση του εργαστηρίου τοπικά προσφέρει αυξημένο έλεγχο και ασφάλεια στα δεδομένα, προάγοντας τη διασφάλιση της ιδιωτικότητας και την ανεξαρτησία από συνδρομητικές υπηρεσίες τρίτων. Τέλος, η υλοποίηση και διαχείριση ενός τέτοιου εργαστηρίου λειτουργεί ως πολύτιμο γνωστικό υπόβαθρο για μελλοντική σταδιοδρομία σε ρόλους διαχειριστή συστημάτων πληροφορικής ή άλλων ειδικοτήτων πληροφορικής.

1.2 Δομή Πτυχιακής

Στο Κεφάλαιο 2 παρουσιάζονται οι τεχνολογίες που θα χρησιμοποιηθούν στην εργασία αρχικά σε θεωρητικό επίπεδο και στην συνέχεια περιγράφονται τα βασικά στοιχεία τους. Στο Κεφάλαιο 3 αναπτύσσεται η σχεδίαση και αρχιτεκτονική της υποδομής που θα υλοποιηθεί, κατά την οποία αναφέρονται σημαντικές πληροφορίες για τα τεχνικά χαρακτηριστικά τις υποδομής καθώς και την επιλογή λογισμικού. Στο Κεφάλαιο 4 παρουσιάζεται αναλυτικά η διαδικασία υλοποίησης του εργαστηρίου από την αρχή έως το τέλος. Ακολουθεί το Κεφάλαιο 5 στο οποίο αναφέρονται οι τρόποι λήψης και επαναφοράς ενός αντίγραφου ασφάλειας και στατιστικά για την χρήση πόρων του εργαστηρίου. Στην συνέχεια ακολουθεί το Κεφάλαιο 6 στο οποίο παρουσιάζω αναλυτικά το προσωπικό μου homelab καθώς και τις υπηρεσίες που φιλοξενώ σε αυτό. Τέλος, ακολουθεί το Κεφάλαιο 7 με τα συμπεράσματα, η βιβλιογραφία στο Κεφάλαιο 8 και στην συνέχεια στο Κεφάλαιο 9 τα παραρτήματα.

1.3 Ο όρος Homelab

Ο όρος homelab αναφέρεται σε ένα προσωπικό εργαστήριο πληροφορικής, δηλαδή σε μια ή περισσότερες συσκευές (υπολογιστές, server, δικτυακό εξοπλισμό) που ο χρήστης εγκαθιστά και διαχειρίζεται στον χώρο του σπιτιού του, με σκοπό την πειραματική λειτουργία υπηρεσιών ή την εκμάθηση νέων τεχνολογιών. Το homelab μπορεί να περιλαμβάνει απλές πλατφόρμες όπως ένα SBC (Raspberry Pi κ.λπ.) , αλλά και ένα σύνθετο σύνολο μηχανημάτων σε διαμόρφωση υψηλής διαθεσιμότητας το οποίο εξυπηρετεί μεγαλύτερο αριθμό υπηρεσιών. Η έννοια του homelab γεννήθηκε από την ανάγκη επαγγελματιών IT να πειραματίζονται με δικτυακές και συστημικές τεχνικές εκτός παραγωγικού περιβάλλοντος (Campanale, 2025). Μέσα από ένα homelab εργαστήριο ο χρήστης μπορεί να κάνει πρακτική εξάσκηση σε τομείς που ενδιαφέρεται, να αναπτύσσει και να δοκιμάζει εφαρμογές και να πειραματίζεται με νέα εργαλεία. Τέλος, το κόστος για να χτίσει κάποιος ένα homelab εργαστήριο μπορεί να ξεκινήσει αρκετά χαμηλά χρησιμοποιώντας παλιό εξοπλισμό που ίσως ήδη κατέχει.

2

Τεχνολογίες Εικονικοποίησης, Κοντέινερ και Ενορχήστρωσης

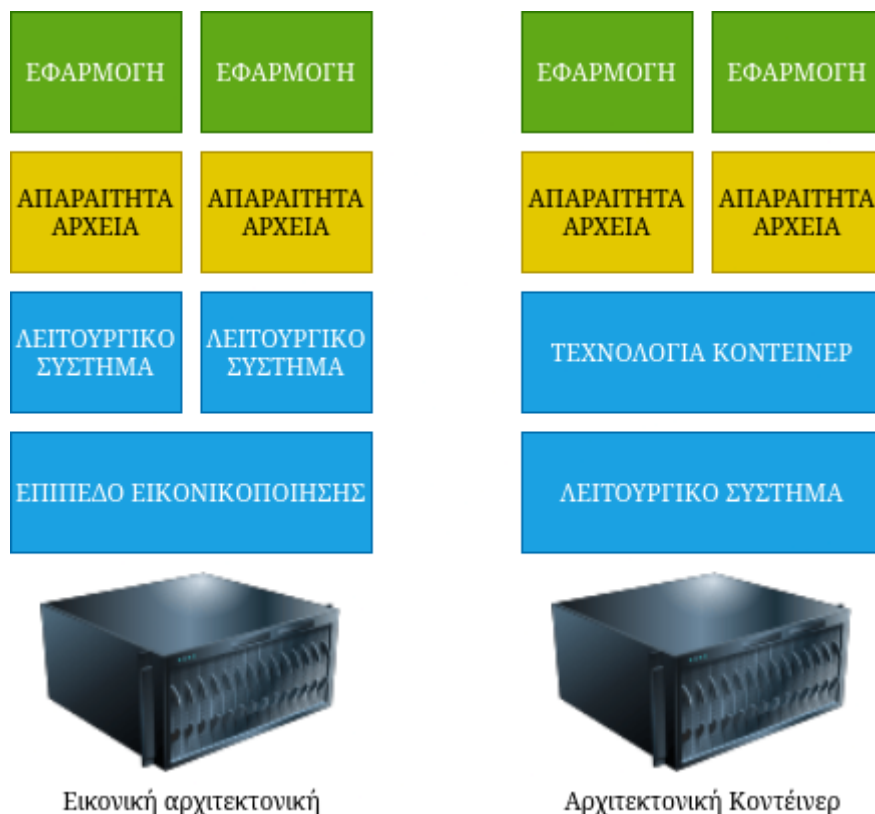
2.1 Τεχνολογίες Εικονικοποίησης (Virtualization)



Εικόνα 1 Διάφορες μεταξύ της παραδοσιακής αρχιτεκτονικής και της εικονικής αρχιτεκτονικής.

Η εικονικοποίηση επιτρέπει σε έναν φυσικό υπολογιστή να «φιλοξενήσει» πολλαπλά ανεξάρτητα λειτουργικά συστήματα (Golden, 2007). Ας φανταστούμε έναν παραδοσιακό υπολογιστή που τρέχει μόνο ένα λειτουργικό σύστημα, π.χ. Windows ή Linux. Με την εικονικοποίηση, μπορούμε να «μοιράσουμε» τον ίδιο υπολογιστή σε πολλά διαφορετικά εικονικά συστήματα, σαν να είχαμε πολλούς ξεχωριστούς υπολογιστές μέσα σε έναν. Επίσης, λόγω του ότι τα λειτουργικά συστήματα φιλοξενούνται σε ένα εικονικό περιβάλλον μας δίνονται αρκετές δυνατότητες όπως η μεταφορά τους από ένα σύστημα σε ένα άλλο χωρίς το εκάστοτε λειτουργικό να δει κάποια διαφορά στο υλικό, αφού είναι εικονικό και παραμένει σταθερό. Άλλη μια δυνατότητα που μας παρέχεται χρησιμοποιώντας την εικονικοποίηση, είναι η κλωνοποίηση μιας εικονικής μηχανής σε ένα αντίγραφο 1:1, αφού όλο το υλικό είναι πρακτικά, αρχεία στο σύστημα μας. Αυτό μας βοηθά στην περίπτωση που θέλουμε να κάνουμε δοκιμές χωρίς να επηρεάσουμε την αρχική εικονική μηχανή. Πλέον, η συγκεκριμένη τεχνολογία χρησιμοποιείται από μικρές και μεγάλες εταιρίες ή ακόμη και σε προσωπικούς ηλεκτρονικούς υπολογιστές, για μεγαλύτερη αποδοτικότητα, μείωση στην έκταση του φυσικού χώρου που καταλαμβάνουν οι φυσικές υπολογιστικές υποδομές, καθώς και στη μείωση των ρίπων λόγω της μειωμένης κατανάλωσης ηλεκτρικής ενέργειας.

2.2 Βασικές Αρχές των Κοντέινερ (Containerization)



Εικόνα 2 Διάφορες μεταξύ της εικονικής αρχιτεκτονικής και της αρχιτεκτονικής κοντέινερ.

Ένα κοντέινερ είναι ένα «πακέτο» που περιέχει μια εφαρμογή και όλα τα αρχεία που χρειάζεται για να λειτουργήσει σωστά. Αντί να εγκαθιστούμε ένα πρόγραμμα απευθείας στο λειτουργικό μας σύστημα, το εγκαθιστούμε μέσα σε ένα κοντέινερ, το οποίο το απομονώνει από το υπόλοιπο σύστημα. Αυτό κάνει τις εφαρμογές πιο ασφαλείς και εύκολες στη διαχείριση και επιπλέον δίνει την δυνατότητα στις εφαρμογές να μπορούν να λειτουργήσουν σε διαφορετικά λειτουργικά συστήματα και αρχιτεκτονικές επεξεργαστών. Τα κοντέινερ απομονώνονται σε επίπεδο διεργασιών μέσα από κοινή χρήση του πυρήνα του κεντρικού λειτουργικού, προσφέροντας πολύ ελαφρύτερα περιβάλλοντα, ταχύτερη εκκίνηση και καλύτερη αξιοποίηση πόρων, με ελαφρώς χαμηλότερο βαθμό απομόνωσης και εξάρτηση από το ίδιο λειτουργικό σύστημα. Αντίθετα, η εικονικοποίηση δημιουργεί πλήρεις εικονικές μηχανές με το δικό τους λειτουργικό σύστημα, παρέχοντας ισχυρή απομόνωση αλλά με μεγαλύτερη κατανάλωση πόρων και πιο αργή εκκίνηση (Lowe, 2017).

2.3 Kubernetes και Ενορχήστρωση Κοντέινερ (Orchestration)

Το Kubernetes (συνχά αναφέρεται ως K8s), γνωστό και ως κυβερνήτης, είναι ένα σύστημα ανοικτού κώδικα ενορχήστρωσης κοντέινερ, σχεδιασμένο για μεγαλύτερες και πιο πολύπλοκες εγκαταστάσεις. Εστιάζει στην αυτοματοποίηση της διάταξης, κλιμάκωσης και διαχείρισης εφαρμογών που βασίζονται σε κοντέινερ. Ενώ το Docker μας επιτρέπει να τρέχουμε μεμονωμένα κοντέινερ, το Kubernetes μας επιτρέπει να τα οργανώνουμε, να τα αναπτύσσουμε αυτόματα και να διαχειριζόμαστε την κατανομή των πόρων τους. Αυτό είναι ιδιαίτερα χρήσιμο όταν έχουμε πολλές υπηρεσίες και θέλουμε να διασφαλίσουμε ότι λειτουργούν συνεχώς χωρίς διακοπές.

Βασικές Λειτουργίες και Πλεονεκτήματα

- Αυτοματοποιημένη Ανάπτυξη (Deployment): Εισαγωγή νέων εκδόσεων εφαρμογών με ελάχιστο downtime.
- Κλιμάκωση (Scaling): Αυτόματη αύξηση ή μείωση του αριθμού των containers ανάλογα με τη ζήτηση και τα στατιστικά απόδοσης.
- Service Discovery & Load Balancing: Αυτόματη ανίχνευση των υπηρεσιών και ισότιμη κατανομή του φορτίου μεταξύ των containers.
- Declarative API: Οι προγραμματιστές ορίζουν μέσω ενός API την επιθυμητή κατάσταση της εφαρμογής και το Kubernetes φροντίζει να διατηρεί το cluster σε αυτήν την κατάσταση.

Αρχιτεκτονική του Kubernetes

- Cluster: Ομάδα από διασκορπισμένους servers (nodes) που συνεργάζονται για την εκτέλεση εφαρμογών.
- Node: Κάθε μηχάνημα (φυσικό ή εικονικό) που εκτελεί containers.
- Pod: Η μικρότερη μονάδα στο Kubernetes, που μπορεί να περιέχει ένα ή περισσότερα containers.
- Control Plane: Το κεντρικό σύστημα που διαχειρίζεται και ελέγχει ολόκληρο το cluster.

3

Σχεδίαση και Αρχιτεκτονική

3.1 Απαιτήσεις Συστήματος και Περιορισμοί

Είναι σημαντικό κατά την σχεδίαση ενός Homelab εργαστήριου να γίνει ένας υπολογισμός των απαιτούμενων πόρων που θα χρησιμοποιηθούν από όλα τα λειτουργικά συστήματα και λογισμικά που σκοπεύουμε να τρέξουμε σε αυτό. Ποιο συγκεκριμένα, θα πρέπει να υπολογίσουμε τους πόρους συστήματος που θα καταναλώνονται σύμφωνα με την σελίδα τεχνικών προδιαγραφών του εκάστοτε λειτουργικού συστήματος ή λογισμικού, όπου αναγράφονται οι απαιτούμενοι πόροι.

Συγκεκριμένα για την παρούσα εργασία οι ελάχιστες και προτεινόμενες απαιτήσεις υλικού υπολογίζονται ως εξής:

Proxmox VE¹:

- Επεξεργαστής: 64-bit Intel ή AMD64 με υποστήριξη Intel VT/AMD-V.
- Μνήμη RAM: Ελάχιστο μέγεθος 2 GB για το λειτουργικό και τις υπηρεσίες Proxmox VE.
- Σκληρός δίσκος: Τύπου SSD με τουλάχιστον 32GB για το λειτουργικό σύστημα.

Ubuntu εικονική μηχανή με εγκατεστημένο το K3s²:

- Επεξεργαστής: Ελάχιστο 2 πυρήνες.
- Μνήμη RAM: Ελάχιστο μέγεθος 2 GB.
- Σκληρός δίσκος: Τύπου SSD με τουλάχιστον 10GB.

Ubuntu LXC κοντέινερ με εγκατεστημένο το Docker³

- Επεξεργαστής: Ελάχιστο 1 πυρήνες.
- Μνήμη RAM: Ελάχιστο μέγεθος 2 GB.
- Σκληρός δίσκος: Τύπου SSD με τουλάχιστον 10GB.

Για να εκτιμηθούν οι πόροι του φυσικού μηχανήματος που θα χρησιμοποιηθεί αθροίζουμε τις επιμέρους απαιτήσεις:

¹ <https://www.proxmox.com/en/products/proxmox-virtual-environment/requirements>

² <https://docs.k3s.io/installation/requirements>

³ <https://docs.docker.com/engine/install/ubuntu/>

- Επεξεργαστής: 2 (VM) + 1 (LXC) + 1–2 (Επιβάρυνση κεντρικού υπολογιστή) = ~4 πυρήνες.
- Μνήμη RAM: 2 GB (Proxmox) + 2 GB (VM) + 2 GB (LXC) + 2 GB (προσωρινή μνήμη λειτουργικού συστήματος) = ~8 GB.
- Σκληρός δίσκος: 64 GB (Proxmox) + 20 GB (VM) + 10 GB (LXC) = ~100 GB SSD.

3.2 Διάταξη Υλικού και Δικτύων

Σε αυτή την εργασία το υλικό που θα χρησιμοποιηθεί θα είναι εικονικό και θα φιλοξενείται στο προσωπικό μου Homelab το οποίο παρουσιάζεται στο Κεφάλαιο 6. Σε ένα πραγματικό σενάριο κάποιος που θα δημιουργήσει ένα Homelab εργαστήριο θα το κάνει χρησιμοποιώντας κάποιο υπολογιστικό σύστημα που έχει στη διάθεση του. Αυτό μπορεί να είναι κάποιο παλιό desktop ή laptop που δεν χρησιμοποιεί πλέον είτε ακόμη και κάποιο SBC όπως το Raspberry Pi κ.λπ.

Ως δίκτυο θα χρησιμοποιήσουμε αυτό που προσφέρεται από το ρούτερ του παρόχου. Στην συγκεκριμένη εργασία είναι το: 10.240.100.0/24, αλλά δεν έχει πολύ σημασία διότι δεν έχει κάποια διαφορά στην λειτουργικότητα με ένα πραγματικό σενάριο.

3.3 Επιλογή Λειτουργικού συστήματος και Λογισμικού

3.3.1 Proxmox Virtual Environment (PVE)

Το Proxmox είναι μια δωρεάν και ανοιχτού κώδικα πλατφόρμα διαχείρισης εικονικοποίησης, η οποία υποστηρίζει μια πληθώρα τεχνολογιών, με τις κυριότερες να είναι KVM (Kernel-based Virtual Machine) και LXC (Linux Containers). Επίσης, παρέχει ένα εύχρηστο διαχειριστικό περιβάλλον μέσω web, καθιστώντας τη διαχείριση του απλή ακόμα και για αρχάριους χρήστες (Tonello, 2022).

Το KVM είναι μια αρκετά διαδεδομένη και δωρεάν τεχνολογία, η οποία είναι μέρος του Linux Kernel που επιτρέπει τη δημιουργία πλήρως απομονωμένων εικονικών μηχανών. Κάθε εικονική μηχανή έχει το δικό της λειτουργικό σύστημα, ανεξάρτητα από το λειτουργικό σύστημα του κεντρικού server. Η τεχνολογία αυτή καταναλώνει ελάχιστους πόρους η ίδια για να λειτουργήσει στο σύστημα, αφήνοντας έτσι περισσότερους πόρους ελεύθερους για το λειτουργικό σύστημα που θέλουμε να τρέξουμε σε αυτό. Μπορεί να προσομοιώνει πολλά λειτουργικά συστήματα, με κυριότερα τα Windows, Linux και Mac OS.

Η τεχνολογία LXC (Linux Containers) είναι μια ελαφρύτερη μορφή εικονικοποίησης, όπου το ίδιο λειτουργικό σύστημα μπορεί να «φιλοξενεί» πολλαπλά απομονωμένα περιβάλλοντα. Σε αντίθεση με το KVM, τα κοντέινερ δεν χρειάζονται πλήρες λειτουργικό σύστημα, αλλά μοιράζονται τον ίδιο Linux kernel του συστήματος. Αυτό τα καθιστά πιο αποδοτικά σε πόρους συστήματος, ιδανικά για ελαφριές εφαρμογές ή δοκιμές.

3.3.2 Docker & Docker Compose

Το Docker είναι ένα λογισμικό, το οποίο μπορούμε να φανταστούμε ως μια εργαλειοθήκη που μας επιτρέπει την εύκολη δημιουργία, εκτέλεση και διαχείριση κοντέινερ. Χρησιμοποιώντας το Docker μπορούμε να δημιουργήσουμε ένα πρότυπο κοντέινερ με την εφαρμογή μας και όλα τα απαραίτητα επιπλέον προγράμματα και βιβλιοθήκες που χρειάζεται για να λειτουργήσει, ή να χρησιμοποιήσουμε έτοιμα πρότυπα τα οποία έχουν δημιουργηθεί από τρίτους. Ένα πρότυπο κοντέινερ Docker μπορεί να εκτελεσθεί μια ή πολλαπλές φορές

μέσα στο ίδιο λειτουργικό σύστημα, χρησιμοποιώντας λιγότερους πόρους συστήματος από μια εικονική μηχανή. Το γεγονός ότι το Docker υποστηρίζεται από πολλά λειτουργικά συστήματα, μας επιτρέπει στα πρότυπα κοντέινερ και κατά συνέπεια στις εφαρμογές τους να λειτουργήσουν ανεξάρτητα του εκάστοτε λειτουργικού συστήματος.

Ο τρόπος δημιουργίας ενός προτύπου κοντέινερ χρησιμοποιώντας το Docker είναι μέσω του Dockerfile. Το Dockerfile είναι ένα αρχείο κειμένου από το οποίο το Docker διαβάζει τις οδηγίες που θα ακολουθήσει για να δημιουργήσει το πρότυπο κοντέινερ. Η σύνταξη των εντολών στο Dockerfile είναι προκαθορισμένη και μια περιγραφή τους μπορεί να βρεθεί στο <https://docs.docker.com/reference/dockerfile/>. Ένα παράδειγμα ενός Dockerfile φαίνεται στην Εικόνα 3 στην οποία απεικονίζονται οι οδηγίες για να δημιουργήσουμε έναν βασικό διακομιστή ιστού Nginx.

```
1 # Χρησιμοποιούμε το επίσημο nginx πρότυπο.
2 FROM nginx:1.28.0
3
4 # Αντιγράφουμε το αρχείο index.html στον κατάλληλο φάκελο του nginx μέσα στο κοντέινερ.
5 COPY index.html /usr/share/nginx/html/index.html
```

Εικόνα 3 Παράδειγμα ενός Dockerfile το οποίο δημιουργεί έναν διακομιστή ιστού nginx.

- FROM nginx:1.28.0

Αυτή η γραμμή, λέει στο Docker να χρησιμοποιήσει ως βάση το επίσημο nginx image, στην έκδοση 1.28.0.

- COPY index.html /usr/share/nginx/html/index.html

Με αυτή την γραμμή, αντιγράφουμε το τοπικό αρχείο index.html στον κατάλληλο φάκελο εντός του κοντέινερ. Αυτό θα είναι η αρχική σελίδα που θα φορτώνει ο nginx όταν επισκεπτόμαστε την σελίδα από έναν περιηγητή ιστού.

Το Docker Compose είναι ένα εργαλείο που μας επιτρέπει να ορίζουμε σε μια προκαθορισμένη μορφή, πολλαπλά πρότυπα Docker κοντέινερ καθώς και όλες τις απαραίτητες ρυθμίσεις που χρειάζονται για να λειτουργήσουν, σε ένα αρχείο ρυθμίσεων τύπου YAML. Αν έχουμε μια ιστοσελίδα και μια βάση δεδομένων, η κάθε μία τρέχει σε δικό της κοντέινερ. Το Docker Compose μας βοηθάει να τα συνδέσουμε όλα μαζί και να τα τρέξουμε εύκολα με ένα αρχείο YAML. Έτσι, δεν χρειάζεται να τρέξουμε πολλαπλές εντολές, αλλά μόνο μία για να ξεκινήσουμε όλη την εφαρμογή. Επίσης, το Docker Compose επιτρέπει λόγω του ότι αποθηκεύει όλες τις απαραίτητες προδιαγραφές και οδηγίες για την εκκίνηση των κοντέινερ σε ένα αρχείο, τη μεταφορά του αρχείου YAML σε έναν άλλο υπολογιστή έτσι ώστε να εκκινηθούν ακριβώς τα ίδια κοντέινερ.

3.3.3 K3s Kubernetes

Το K3s είναι ένα εργαλείο που επιτρέπει την εγκατάσταση του Kubernetes χρησιμοποιώντας ελάχιστους πόρους σε σχέση με την παραδοσιακή έκδοση του Kubernetes. Επίσης, αυτοματοποιεί την διαδικασία εγκατάστασης και επιτρέπει την εγκατάσταση μέσα σε μερικά λεπτά αντί του παραδοσιακού τρόπου που θα χρειαζόταν αρκετές ώρες. Το K3s παρέχεται ως ένα ενιαίο εκτελέσιμο αρχείο μικρότερο των 100MB, που περιλαμβάνει όλα τα απαραίτητα components για τη λειτουργία ενός cluster. Η εγκατάσταση ενός cluster μπορεί να γίνει με μία μόνο εντολή, χωρίς περίπλοκες εξαρτήσεις ή ρυθμίσεις. Δεν απαιτείται ξεχωριστή εγκατάσταση πολλών υπηρεσιών όπως στο παραδοσιακό Kubernetes.

3.3.4 Gatus

Το Gatus είναι ένα λογισμικό ανοιχτού κώδικα το οποίο έχει σχεδιαστεί για την παρακολούθηση λειτουργικότητας εφαρμογών. Μας επιτρέπει να ορίσουμε για κάθε εφαρμογή τα κριτήρια τα οποία την καθιστούν λειτουργική, δηλαδή ελέγχους υγείας οι οποίοι μπορεί να είναι απλοί αλλά και σύνθετοι. Επίσης, το Gatus προσφέρει την δυνατότητα ειδοποιήσεων μέσω πολλών γνωστών πρωτοκόλλων όπως μηνύματα ηλεκτρονικού ταχυδρομείου, σε περίπτωση που μια εφαρμογή δεν πληροί τους ελέγχους υγείας. Για παράδειγμα μπορούμε να ορίσουμε ένα κριτήριο για την λήξη του πιστοποιητικού SSL της ιστοσελίδα μας ώστε, για παράδειγμα στις 7 ημέρες πριν την λήξη του να λαμβάνουμε ειδοποίηση ώστε να το ανανεώσουμε. Τέλος, η εγκατάσταση του είναι αρκετά εύκολη, με προτεινόμενο τρόπο εγκατάστασης σε Docker κοντέινερ.

3.4 Ζητήματα Ασφαλείας

Κατά τον σχεδιασμό ενός Homelab εργαστήριου είναι σημαντικό να σχεδιαστεί καλύπτοντας κάποια βασικά ζητήματα ασφάλειας που μπορούμε να προκαθορίσουμε, δημιουργώντας κανόνες και μεθόδους που μας βοηθούν να το πέτυχουμε. Ένα βασικό ζήτημα ασφάλειας σε κάθε σύστημα, φυσικό ή εικονικό, είναι το τείχος προστασίας και η άμυνα προς άγνωστες διευθύνσεις που δεν χρειάζεται να έχουν πρόσβαση στο εκάστοτε σύστημα. Μια καλή πρακτική σε ένα τείχος προστασίας, είναι να επιτρέπουμε την εισερχόμενη κίνηση μόνο από γνωστές διευθύνσεις ή δίκτυα τα οποία θεωρούμε έμπιστα, ακόμη και αν βρισκόμαστε σε ένα κλειστό περιβάλλον όπως για παράδειγμα ένα Homelab σε ένα σπίτι. Ακόμη, μια καλή πρακτική είναι η συχνή και τακτική ενημέρωση των συστημάτων στην τελευταία έκδοση, ώστε να αποφεύγονται πιθανές ευπάθειες και προβλήματα συστήματος. Σημαντικό επίσης, είναι τα συχνά αντίγραφα ασφαλείας τόσο των ρυθμίσεων φυσικών συστημάτων, καθώς και ολοκλήρων των εικονικών μηχανών και κοντέινερ που χρησιμοποιούμε ώστε να αποφύγουμε τυχόν απώλεια δεδομένων.

4

Υλοποίηση

4.1 Εγκατάσταση και Ρύθμιση του Proxmox VE

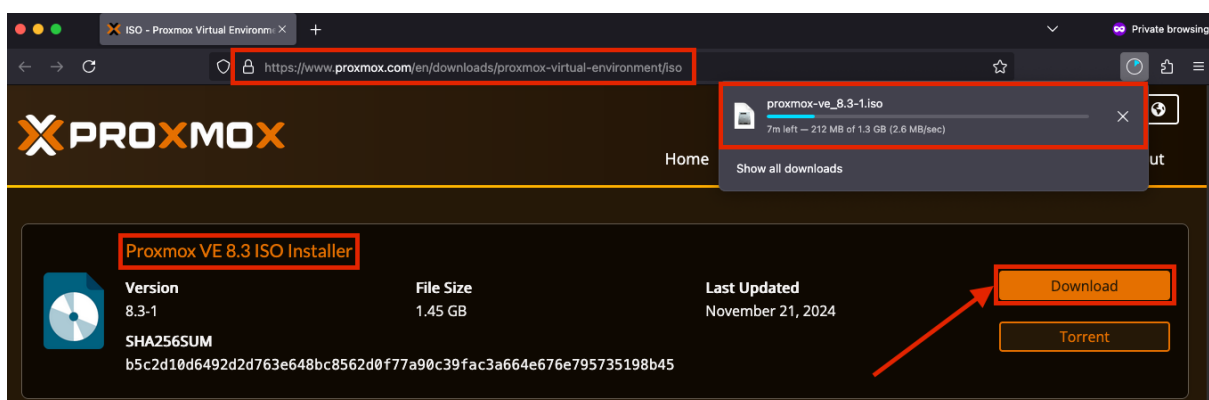
Πριν ξεκινήσουμε την εγκατάσταση του Proxmox πρέπει να σιγουρευτούμε ότι το υλικό που έχουμε επιλέξει στο οποίο θα το εγκαταστήσουμε, πληροί όλες τις απαιτήσεις μας όπως αναφέρεται στο υποκεφάλαιο 3.1. Για την εργασία αυτή θα χρησιμοποιηθεί μια εικονική μηχανή η οποία θα προσομοιώνει έναν ηλεκτρονικό υπολογιστή με τα εξής χαρακτηριστικά:

- Επεξεργαστής με 4 πυρήνες
- Μνήμη RAM 8GB
- Σκληρός δίσκος 256GB SSD

Η διαδικασία παρόλο που στην προκείμενη περίπτωση ακολουθείται σε μια εικονική μηχανή, είναι ίδια με τη διαδικασία που θα ακολουθούσαμε αν χρησιμοποιούνταν ένας φυσικός ηλεκτρονικός υπολογιστής. Τα βήματα που απαιτούνται για την εγκατάσταση του Proxmox είναι τα εξής:

4.1.1 Λήψη του είδωλου εγκατάστασης (ISO)

Αρχικά θα πρέπει να επισκεφτούμε την επίσημη ιστοσελίδα του Proxmox⁴, και από την λίστα προϊόντων να επιλέξουμε την τελευταία έκδοση του Proxmox VE ISO Installer. Κατά την εγγραφή της παρούσας εργασίας η τελευταία έκδοση είναι 8.3, συνεπώς επιλέξαμε το Proxmox VE 8.3 ISO Installer. Πατώντας το κουμπί «Download» ξεκινάει η λήψη του όπως φαίνεται στην Εικόνα 4.



Εικόνα 4 Λήψη του είδωλου εγκατάστασης (ISO) από την επίσημη ιστοσελίδα του Proxmox.

⁴ <https://www.proxmox.com/en/downloads/proxmox-virtual-environment/iso>

4.1.2 Εγγραφή του είδωλού εγκατάστασης (ISO) σε μονάδα USB

Η διαδικασία εγγραφής του είδωλού εγκατάστασης (ISO) σε μονάδα USB διαφέρει ανάλογα με το λογισμικό του υπολογιστή που θα χρησιμοποιήσουμε. Παρακάτω αναφέρονται εν συντομία τα βήματα για τα λειτουργικά Linux, Windows και MacOS. Στην δική μας περίπτωση χρησιμοποιήσαμε Fedora Linux (Proxmox Prepare Installation Media, 2023).

 Προσοχή! Βεβαιωθείτε ότι η μονάδα δεν περιέχει σημαντικά δεδομένα διότι η παρακάτω διαδικασία θα τα διαγράψει.

Διαδικασία σε Linux

Λόγω του ότι οι εκδόσεις των Linux μπορεί να διαφέρουν αρκετά μεταξύ τους, στο παρακάτω παράδειγμα χρησιμοποιούμε Linux Fedora και χρησιμοποιήσουμε λογισμικά που έρχονται προ εγκατεστημένα στις περισσότερες εκδόσεις. Αρχικά θα πρέπει να εντοπιστεί το όνομα της μονάδας USB συγκρίνοντας το αποτέλεσμα της εντολής «lsblk» πριν και μετά την σύνδεση της μονάδας USB στον υπολογιστή μας. Συνεπώς σε ένα τερματικό εκτελούμε την εντολή «lsblk» και στην συνέχεια συνδέουμε την μονάδα USB και ξανά εκτελούμε την εντολή «lsblk». Συγκρίνοντας αυτές τις 2 εξόδους παρατηρούμε ότι στην δεύτερη εκτέλεση έχει εμφανιστεί μια νέα συσκευή όπως φαίνεται στην Εικόνα 5. Παρατηρούμε επίσης ότι στην στήλη «TYPE» έχει την τιμή «disk» οπότε στην στήλη «NAME» βρίσκεται και το όνομα που ψάχνουμε. Για σιγουριά ελέγχουμε ότι το μέγεθος στην στήλη «SIZE» ταιριάζει με το αναγραφόμενο μέγεθος της μονάδας USB. Στην συνέχεια θα χρησιμοποιήσουμε την εντολή “dd” ώστε να αντιγράψουμε το είδωλο εγκατάστασης (ISO) στην μονάδα USB (η εντολή θα πρέπει να εκτελεστεί μέσα από τον φάκελο όπου έχουμε κάνει λήψη του ειδώλου εγκατάστασης (ISO)):

```
# dd bs=1M conv=fdatasync if=./proxmox-ve_*.iso of=/dev/<μονάδα USB>
```

Επεξήγηση εντολής:

- **dd:** Είναι ένα εργαλείο στα Linux για αντιγραφή και μετατροπή δεδομένων σε χαμηλό επίπεδο. Χρησιμοποιείται συχνά για να αντιγράψει αρχεία ή ολόκληρες συσκευές, όπως USB, διατηρώντας όλα τα δεδομένα.
- **bs=1M:** Ορίζει το μέγεθος του block (π.χ. 1 Megabyte) που αντιγράφεται κάθε φορά. Αυτό επιταχύνει την διαδικασία αντιγραφής σε σχέση με μικρότερα blocks.
- **conv=fdatasync:** Αναγκάζει το σύστημα να γράψει όλα τα δεδομένα πριν τελειώσει η εντολή. Αυτό εξασφαλίζει ότι όλα τα δεδομένα έχουν αποθηκευτεί σωστά στην μονάδα USB και δεν παραμένουν στη προσωρινή μνήμη.
- **if=./proxmox-ve_*.iso:** Το αρχείο προέλευσης (input file). Σε αυτή την περίπτωση, είναι το ISO αρχείο του Proxmox. Το «*» σημαίνει ότι θα βρεθεί το πρώτο αρχείο που ταιριάζει με το μοτίβο του ονόματος.
- **of=/dev/<μονάδα USB>:** Ο προορισμός (output file). Είναι η μονάδα USB (π.χ. /dev/sda).

Όπου <μονάδα USB> είναι το όνομα που πήραμε από την στήλη «NAME» παραπάνω. Στην Εικόνα 5 φαίνεται επίσης η εντολή που τρέξαμε στην περίπτωση μας. Αφού ολοκληρωθεί η διαδικασία μπορούμε να αφαιρέσουμε την μονάδα USB από τον υπολογιστή μας και να ξεκινήσουμε με την εγκατάσταση του Proxmox.

```

:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
zram0                               252:0    0   8G  0 disk  [SWAP]
nvme0n1                             259:0    0 476.9G  0 disk
├─nvme0n1p1                         259:1    0   600M  0 part  /boot/efi
├─nvme0n1p2                         259:2    0    1G  0 part  /boot
└─nvme0n1p3                         259:3    0 475.4G  0 part
   └─luks-                           253:0    0 475.3G  0 crypt  /home
                                     /

:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda                                 8:0     1 28.7G  0 disk
├─sda1                             8:1     1 28.6G  0 part  /run/media
└─sda2                             8:2     1   32M  0 part
zram0                               252:0    0   8G  0 disk  [SWAP]
nvme0n1                             259:0    0 476.9G  0 disk
├─nvme0n1p1                         259:1    0   600M  0 part  /boot/efi
├─nvme0n1p2                         259:2    0    1G  0 part  /boot
└─nvme0n1p3                         259:3    0 475.4G  0 part
   └─luks-                           253:0    0 475.3G  0 crypt  /home
                                     /

:~$ dd bs=1M conv=fdatasync if=./proxmox-ve_*.iso of=/dev/sda

```

Εικόνα 5 Διαδικασία εύρεσης της μονάδας USB και εγγραφή του ειδώλου εγκατάστασης (ISO) σε αυτή στα Linux.

Διαδικασία σε Windows και MacOS

Για τα Windows και MacOS μπορούμε να χρησιμοποιήσουμε μια δωρεάν εφαρμογή, την balenaEtcher, ακολουθώντας τις οδηγίες στην επίσημη σελίδα <https://etcher.balena.io/> για την εγκατάσταση και τις οδηγίες χρήσης του.

4.1.3 Εγκατάσταση του Proxmox

Αφού έχουμε ολοκληρώσει την εγγραφή του ειδώλου εγκατάστασης στην μονάδα USB, θα πρέπει να ανατρέξουμε στο εκάστοτε εγχειρίδιο του ηλεκτρονικού υπολογιστή που θα το εγκαταστήσουμε, ώστε να βρούμε τον τρόπο με τον οποίο επιτρέπει την εκκίνηση από την μονάδα USB. Αυτό διαφέρει ανάλογα το μοντέλο του ηλεκτρονικού υπολογιστή. Στην περίπτωση μας έπρεπε να πατήσουμε το κουμπί F11 στο πληκτρολόγιο ώστε ο υπολογιστής να εκκινήσει τον οδηγό εγκατάστασης του Proxmox που βρίσκεται στην μονάδα USB. Στο μενού του οδηγού εγκατάστασης που φαίνεται στην Εικόνα 6 επιλέγουμε «Install Proxmox VE (Graphical)».

Proxmox VE 8.3 (iso release 1) - <https://www.proxmox.com/>



Welcome to Proxmox Virtual Environment

Install Proxmox VE (Graphical)
Install Proxmox VE (Terminal UI)
Advanced Options

enter: select, arrow keys: navigate, e: edit entry, esc: back

Εικόνα 6 Οδηγός εγκατάστασης του Proxmox.

Αμέσως μετά θα ξεκινήσει η φόρτωση των απαραίτητων αρχείων για την εγκατάσταση και θα εμφανίσει ο οδηγός εγκατάστασης όπως φαίνεται στην Εικόνα 7. Αφού διαβάσουμε την συμφωνία αδειοδότησης τελικού-χρήστη που αναγράφεται και εφόσον συμφωνούμε, συνεχίζουμε πατώντας το κουμπί «I agree».

END USER LICENSE AGREEMENT (EULA)

END USER LICENSE AGREEMENT (EULA) FOR PROXMOX VIRTUAL ENVIRONMENT (PROXMOX VE)

By using Proxmox VE software you agree that you accept this EULA, and that you have read and understand the terms and conditions. This also applies for individuals acting on behalf of entities. This EULA does not provide any rights to Support Subscriptions Services as software maintenance, updates and support. Please review the Support Subscriptions Agreements for these terms and conditions. The EULA applies to any version of Proxmox VE and any related update, source code and structure (the Programs), regardless of the delivery mechanism.

1. License. Proxmox Server Solutions GmbH (Proxmox) grants to you a perpetual, worldwide license to the Programs pursuant to the GNU Affero General Public License V3. The license agreement for each component is located in the software component's source code and permits you to run, copy, modify, and redistribute the software component (certain obligations in some cases), both in source code and binary code forms, with the exception of certain binary only firmware components and the Proxmox images (e.g. Proxmox logo). The license rights for the binary only firmware components are located within the components. This EULA pertains solely to the Programs and does not limit your rights under, or grant you rights that supersede, the license terms of any particular component.

2. Limited Warranty. The Programs and the components are provided and licensed "as is" without warranty of any kind, expressed or implied, including the implied warranties of merchantability, non-infringement or fitness for a particular purpose. Neither Proxmox nor its affiliates warrants that the functions contained in the Programs will meet your requirements or that the operation of the Programs will be entirely error free, appear or perform precisely as described in the accompanying documentation, or comply with regulatory requirements.

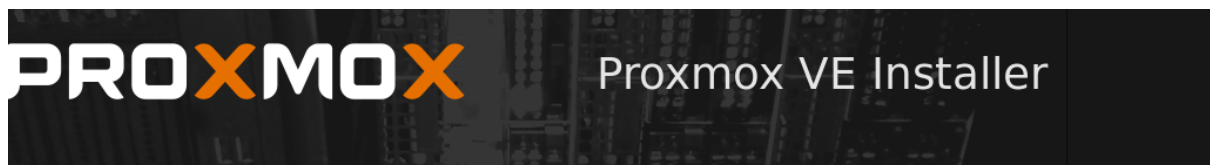
3. Limitation of Liability. To the maximum extent permitted under applicable law, under no circumstances will Proxmox, its affiliates, any Proxmox authorized distributor, or the licensor

Previous

I agree

Εικόνα 7 Συμφωνία αδειοδότησης τελικού-χρήστη Proxmox.

Στην συνέχεια θα πρέπει να επιλέξουμε τον σκληρό δίσκο στον οποίο θα εγκαταστήσουμε το Proxmox. Στην περίπτωση που δεν έχουμε πάνω από έναν δίσκο, ο οδηγός θα επιλέξει αυτόματα τον δίσκο. Παρόλα αυτά θα πρέπει να ελέγξουμε και να επιβεβαιώσουμε ότι έχει επιλεγεί ο δίσκος που επιθυμούμε, διότι ο οδηγός θα διαγράψει όλα τα αρχεία από αυτόν. Στην συνέχεια πατάμε ο κουμπί «Next» όπως φαίνεται στην Εικόνα 8.

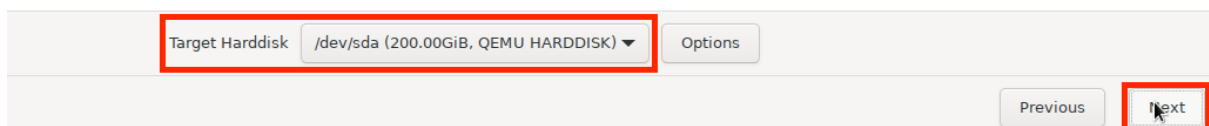


Proxmox Virtual Environment (PVE)

The Proxmox Installer automatically partitions your hard disk. It installs all required packages and makes the system bootable from the hard disk. All existing partitions and data will be lost.

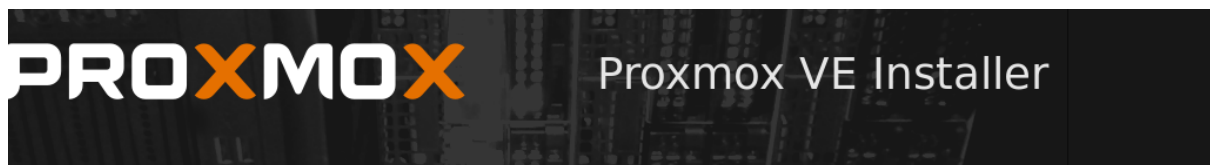
Press the Next button to continue the installation.

- **Please verify the installation target**
The displayed hard disk will be used for the installation.
Warning: All existing partitions and data will be lost.
- **Automatic hardware detection**
The installer automatically configures your hardware.
- **Graphical user interface**
Final configuration will be done on the graphical user interface, via a web browser.



Εικόνα 8 Επιλογή σκληρού δίσκου κατά την εγκατάσταση του Proxmox.

Στην επόμενη οθόνη θα επιλέγουμε την τοποθεσία και ζώνη ώρας που επιθυμούμε και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 9.



Location and Time Zone selection

The Proxmox Installer automatically makes location-based optimizations, like choosing the nearest mirror to download files from. Also make sure to select the correct time zone and keyboard layout.

Press the Next button to continue the installation.

- **Country:** The selected country is used to choose nearby mirror servers. This will speed up downloads and make updates more reliable.
- **Time Zone:** Automatically adjust daylight saving time.
- **Keyboard Layout:** Choose your keyboard layout.

Country: Greece

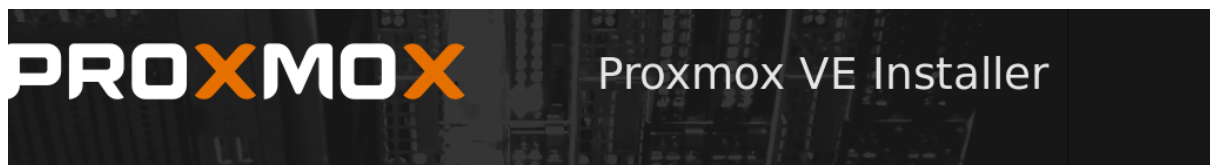
Time zone: Europe/Athens

Keyboard Layout: U.S. English

Previous Next

Εικόνα 9 Επιλογή ρυθμίσεων τοποθεσίας και ζώνης ώρας κατά την εγκατάσταση του Proxmox.

Στην επόμενη οθόνη όπως φαίνεται στην Εικόνα 10, ο οδηγός μας ζητά να εισάγουμε τον κεντρικό κωδικό πρόσβασης του κεντρικού διαχειριστή καθώς και μια διεύθυνση ηλεκτρονικού ταχυδρομείου, όπου εκεί θα λαμβάνουμε ειδοποιήσεις σε περίπτωση προβλήματος στο Proxmox. Συνιστάται να επιλέγουμε δυνατούς και σύνθετους κωδικούς για μεγαλύτερη ασφάλεια. Στην συνέχεια πατάμε το κουμπί «Next».



Administration Password and Email Address

Proxmox Virtual Environment is a full featured, highly secure GNU/Linux system, based on Debian.

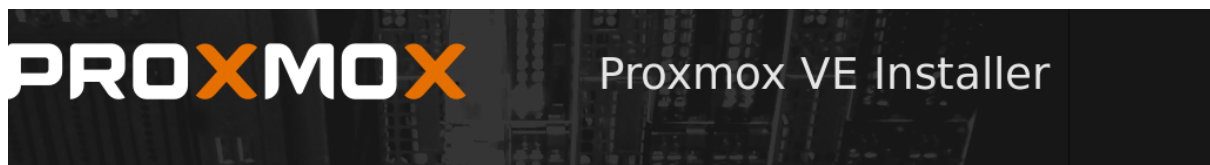
In this step, please provide the `root` password.

- **Password:** Please use a strong password. It should be at least 8 characters long, and contain a combination of letters, numbers, and symbols.
- **Email:** Enter a valid email address. Your Proxmox VE server will send important alert notifications to this email account (such as backup failures, high availability events, etc.).

Press the Next button to continue the installation.

Εικόνα 10 Εισαγωγή διαπιστευτηρίων και διεύθυνση ηλεκτρονικού ταχυδρομείου κατά την εγκατάσταση του Proxmox.

Η επόμενη οθόνη χρειάζεται προσοχή διότι σε αυτή ρυθμίζουμε την κάρτα δικτύου του Proxmox. Αρχικά ο οδηγός μας δίνει μια λίστα στο πεδίο «Management Interface» με κάρτες δικτύου που είναι εγκατεστημένες στο σύστημα μας τις οποίες μπορούμε να επιλέξουμε ποια θα χρησιμοποιήσουμε. Παρατηρούμε ότι υπάρχει μια κουκίδα διπλά από κάθε κάρτα δικτύου στην λίστα, αυτό υποδεικνύει την κατάσταση της. Με κόκκινο συμβολίζεται ότι η κάρτα δικτύου είναι ανενεργή και με πράσινο ότι είναι ενεργοποιημένη και έτοιμη για χρήση. Στην συνέχεια στο πεδίο «Hostname (FQDN)» πρέπει να εισάγουμε ένα πλήρως αναγνωρισμένο όνομα τομέα για το Proxmox. Στην δική μας υποδομή αυτό δεν έχει τόσο μεγάλη σημασία μιας και θα έχουμε πρόσβαση στο Proxmox μέσω της διεύθυνσης IP και όχι του ονόματος του, οπότε επιλέξαμε ως όνομα το «pve.home.lab». Στην συνέχεια ο οδηγός μας έχει συμπληρώσει αυτόματα τα πεδία «IP Address (CIDR)», «Gateway» και «DNS Server» σύμφωνα με τις πληροφορίες που έχει λάβει από τον πάροχό μας. Σε περίπτωση που θέλουμε να αλλάξουμε αυτές τις ρυθμίσεις θα πρέπει να συμβουλευτούμε το εγχειρίδιο χρήσης του δρομολογητή που έχουμε λάβει από τον παροχό μας. Αφού έχουμε ελέγξει και συμπληρώσει όλα τα πεδία πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 11.



Management Network Configuration

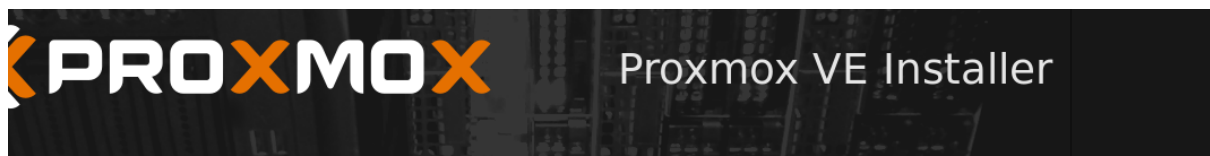
Please verify the displayed network configuration. You will need a valid network configuration to access the management interface after installing.

After you have finished, press the Next button. You will be shown a list of the options that you chose during the previous steps.

- **IP address (CIDR):** Set the main IP address and netmask for your server in CIDR notation.
- **Gateway:** IP address of your gateway or firewall.
- **DNS Server:** IP address of your DNS server.

Εικόνα 11 Επιλογή ρυθμίσεων δικτύου κατά την εγκατάσταση του Proxmox.

Σε αυτό το σημείο ο οδηγός μας προβάλλει μια σύνοψη των επιλογών που έχουμε διαλέξει ώστε να κάνουμε έναν συνολικό έλεγχο πριν την εγκατάσταση. Αφού ελέγξουμε προσεκτικά πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 12.



Summary

Please confirm the displayed information. Once you press the **Install** button, the installer will begin to partition your drive(s) and extract the required files.

Option	Value
Filesystem:	ext4
Disk(s):	/dev/sda
Country:	Greece
Timezone:	Europe/Athens
Keymap:	en-us
Email:	info@home.lab
Management Interface:	enp6s18
Hostname:	pve
IP CIDR:	10.240.100.230/24
Gateway:	10.240.100.1
DNS:	9.9.9.9

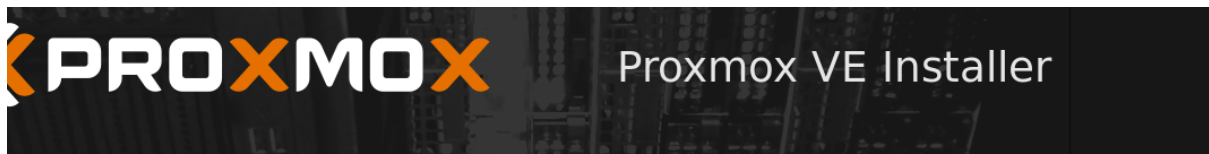
☒ Automatically reboot after successful installation

Previous

Install

Εικόνα 12 Σύνοψη των επιλογών μας πριν την τελική εγκατάσταση του Proxmox.

Τέλος, αφού ολοκληρωθεί με επιτυχία η εγκατάσταση του Proxmox μας εμφανίζει όπως φαίνεται και στην εικόνα 9, την διεύθυνση ιστού στην οποία θα είναι προσβάσιμο το διαχειριστικό περιβάλλον μετά την επανεκκίνηση. Κρατάμε αυτή την διεύθυνση και πατάμε το κουμπί «Reboot». όπως φαίνεται στην Εικόνα 13.



Installation successful!

Proxmox VE is now installed and ready to use.

- **Next steps**

Reboot and point your web browser to the selected IP address on port 8006:

`https://10.240.100.230:8006`

Also visit www.proxmox.com for more information.

Automatic reboot scheduled in 0 seconds.



Reboot

Εικόνα 13 Επιτυχής εγκατάσταση του Proxmox.

Αφού ολοκληρωθεί η επανεκκίνηση, στην οθόνη του ηλεκτρονικού υπολογιστή θα εμφανιστεί η κονσόλα του Proxmox όπως φαίνεται και στην Εικόνα 14. Εκεί αναγράφεται ξανά η διεύθυνση ιστού στην οποία είναι προσβάσιμο το διαχειριστικό περιβάλλον του Proxmox στην περίπτωση που δεν την έχουμε κρατήσει. Σε αυτό το σημείο μπορούμε να αφαιρέσουμε την μονάδα USB από τον ηλεκτρονικό υπολογιστή και να μεταφερθούμε σε έναν άλλο υπολογιστή ώστε να επισκεφτούμε την εν λόγω διεύθυνση ιστού.

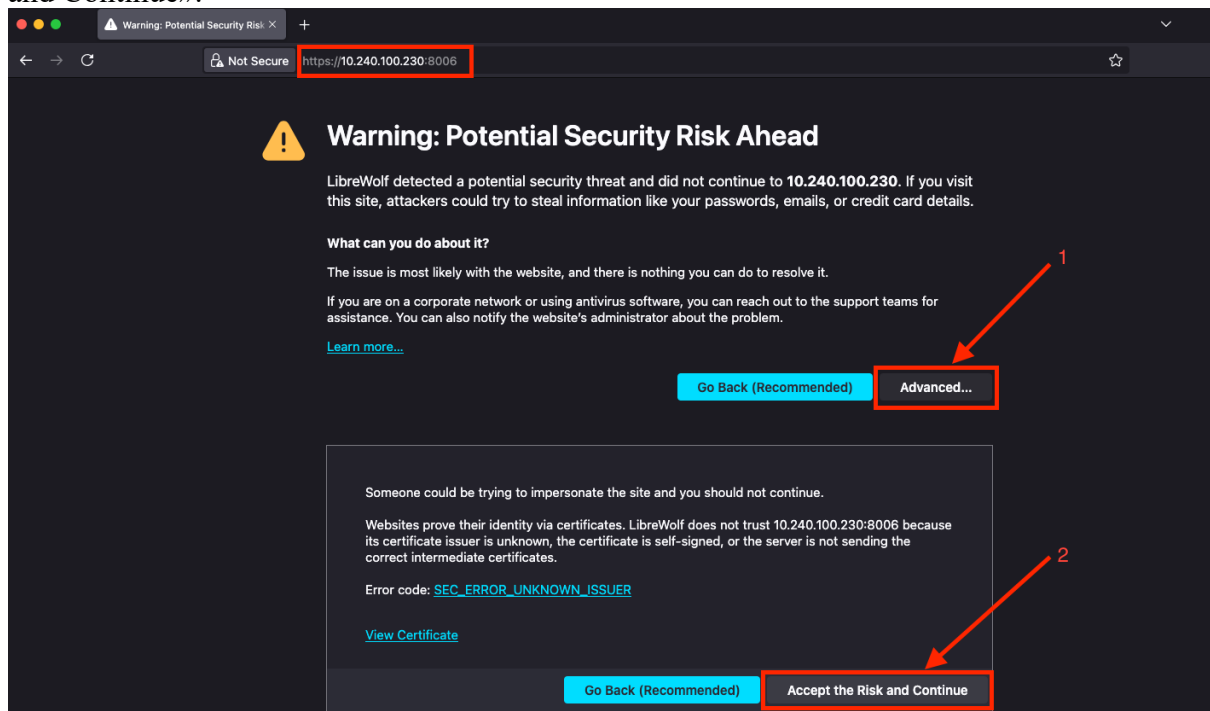


Εικόνα 14 Κονσόλα ελέγχου του Proxmox.

4.1.4 Πρόσβαση στο διαχειριστικό περιβάλλον του Proxmox

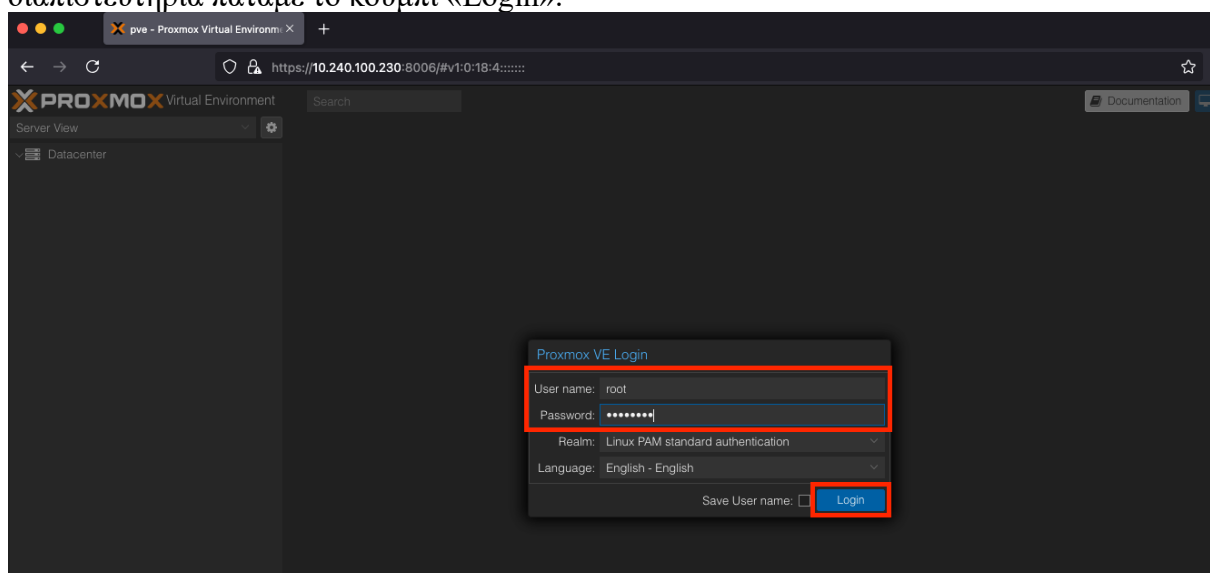
Για να διαχειριστούμε το Promxox θα πρέπει να επισκεφτούμε την διεύθυνση που μας έδωσε ο οδηγός κατά την διάρκεια της εγκατάστασης από έναν περιηγητή ιστού. Στην δική μας περίπτωση «`https://10.240.100.230:8006`». Όπως φαίνεται και στην Εικόνα 15 εμφανίζεται σφάλμα ασφάλειας λόγω του ότι τα πιστοποιητικά που χρησιμοποιεί το Proxmox δεν έχουν

εκδοθεί από κάποιο επίσημο φορέα, αλλά τοπικά κατά την διάρκεια εγκατάστασης, με αποτέλεσμα να μην τα εμπιστεύεται ο περιηγητής. Μπορούμε να αγνοήσουμε αυτό το μήνυμα πατώντας το κουμπί «Advanced...» και στην συνέχεια το κουμπί «Accept the Risk and Continue».



Εικόνα 15 Σφάλμα πιστοποιητικών κατά την πρόσβαση στο διαχειριστικό περιβάλλον του Proxmox.

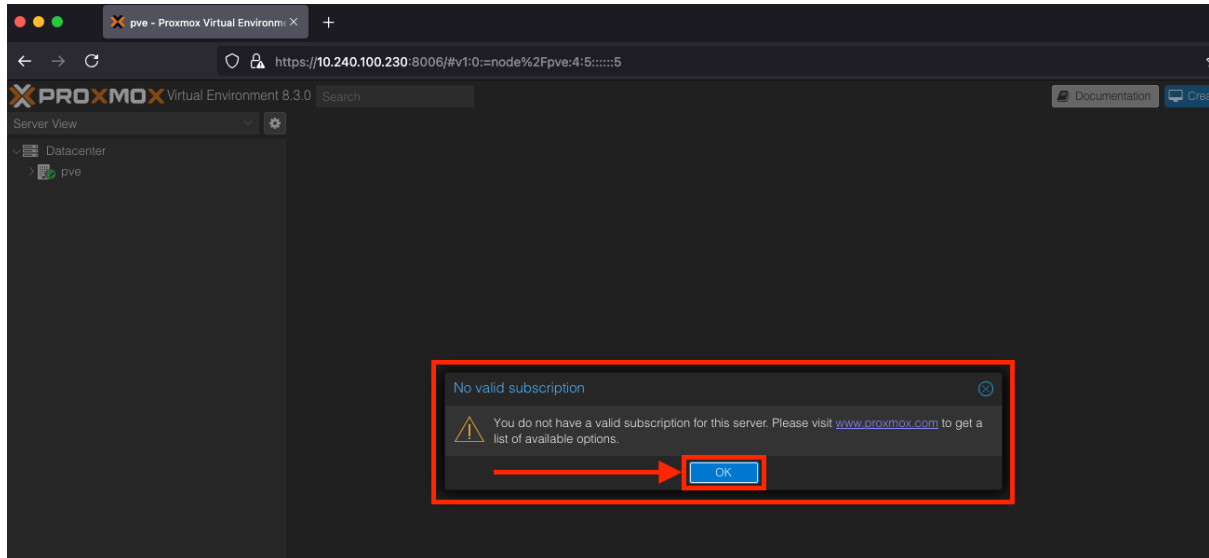
Στην οθόνη πλέον εμφανίζεται το διαχειριστικό περιβάλλον του Proxmox όπως απεικονίζεται και στην Εικόνα 16. Θα πρέπει να κάνουμε σύνδεση χρησιμοποιώντας τα διαπιστευτήρια που δημιουργήσαμε κατά την εγκατάσταση. Ο κεντρικός διαχειριστής πάντα έχει ως όνομα χρήστη (User name) «root» και κωδικό αυτόν που του έχουμε ορίσει. Μόλις εισάγουμε τα διαπιστευτήρια πατάμε το κουμπί «Login».



Εικόνα 16 Είσοδος διαπιστευτηρίων κατά την πρόσβαση στο διαχειριστικό περιβάλλον του Proxmox.

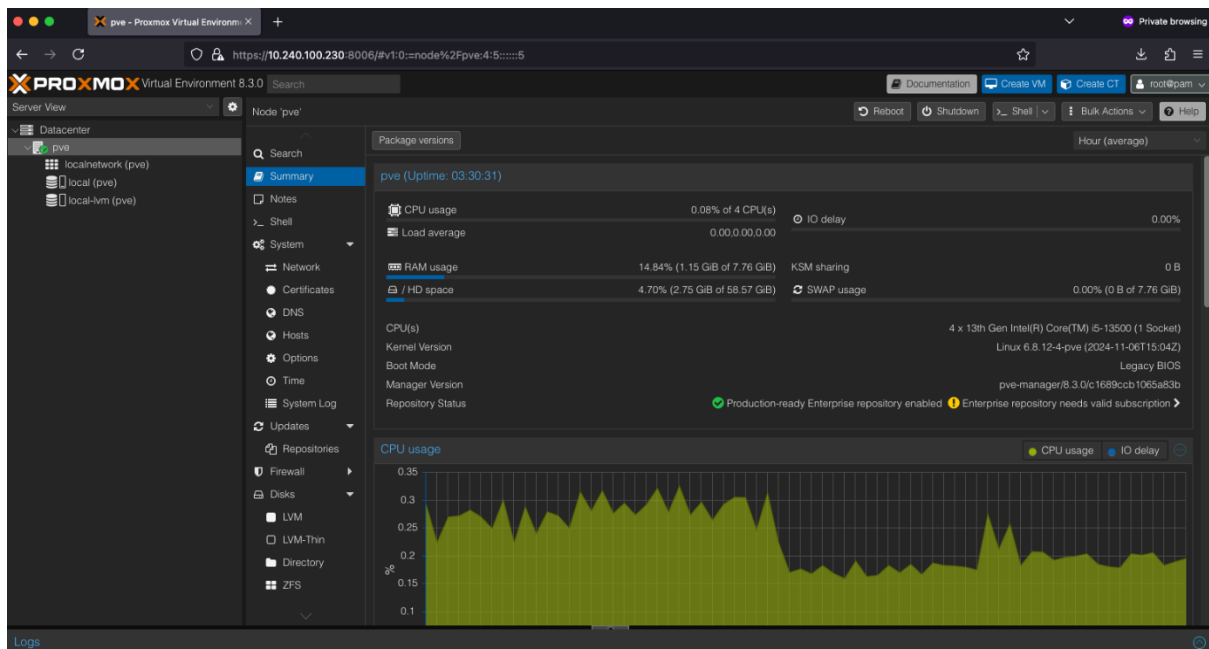
Όταν συνδεθούμε επιτυχώς θα εμφανιστεί το μήνυμα που φαίνεται στην Εικόνα 17, όπου μας προειδοποιεί ότι δεν έχουμε κάποια ενεργή συνδρομή. Αυτό επίσης μπορούμε να το αγνοήσουμε καθ' όλη την διάρκεια χρήσης του διαχειριστικού περιβάλλοντος πατώντας

«OK», μιας και το Proxmox λειτουργεί με το πλήρες πακέτο λειτουργιών με ή χωρίς συνδρομή. Παρόλα αυτά στην περίπτωση μας, επιλέξαμε την χρήση του Proxmox χωρίς συνδρομή οπότε θα πρέπει να κάνουμε κάποιες επιπλέον αλλαγές στις ρυθμίσεις ώστε να ενεργοποιήσουμε τις ενημερώσεις συστήματος όπως εξηγούμε αναλυτικά στη συνέχεια του κειμένου.



Εικόνα 17 Προειδοποίηση για έλλειψη αδείας κατά την πρόσβαση στο διαχειριστικό περιβάλλον του Proxmox.

Τέλος, μας εμφανίζεται το διαχειριστικό περιβάλλον του Proxmox όπως φαίνεται στην παρακάτω Εικόνα 18.

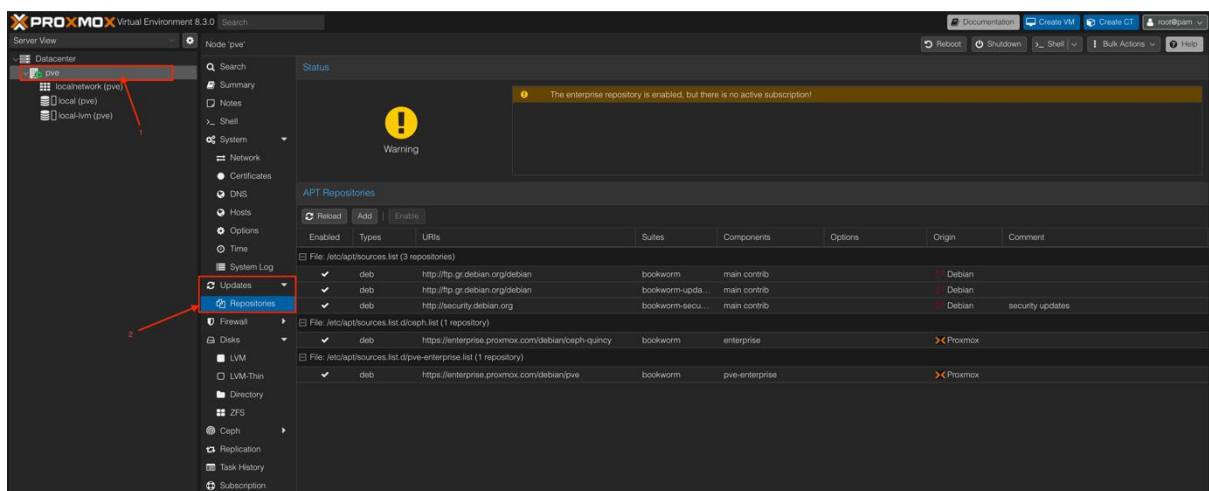


Εικόνα 18 Κεντρική σελίδα διαχειριστικού περιβάλλοντος του Proxmox.

4.1.5 Απαραίτητες ρυθμίσεις μετά την εγκατάσταση του Proxmox

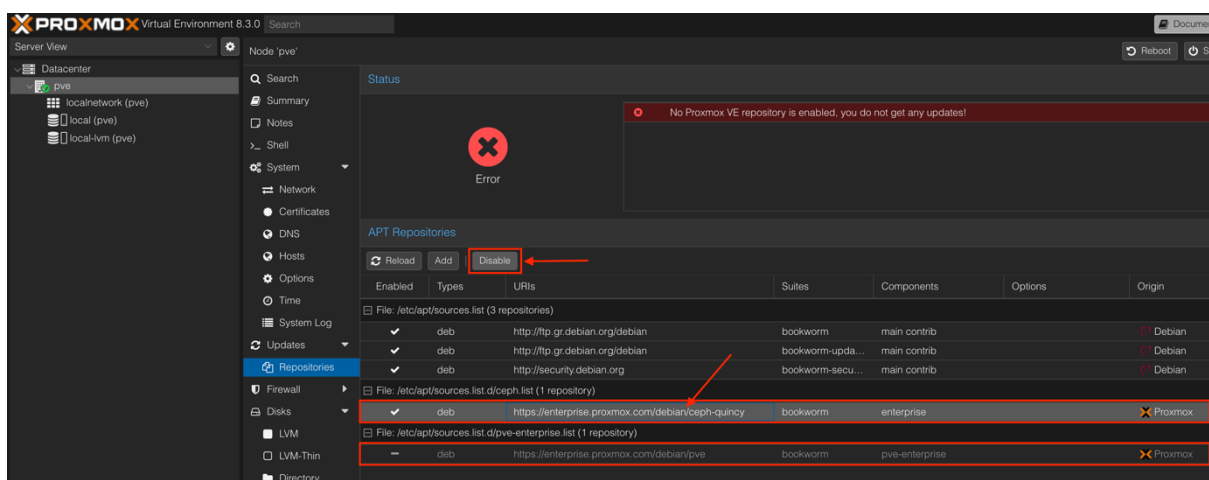
Αφού έχουμε εγκαταστήσει το Proxmox και έχουμε συνδεθεί στο διαχειριστικό του περιβάλλον για πρώτη φορά, θα πρέπει να κάνουμε κάποιες απαραίτητες ρυθμίσεις για να

διασφαλίσουμε την ορθή λειτουργία του. Ποιο συγκεκριμένα, επειδή επιλέξαμε να μην ενεργοποιήσουμε κάποια συνδρομή στο Proxmox, θα πρέπει να αλλάξουμε τις ρυθμίσεις των ενημερώσεων συστήματος ώστε να λειτουργούν εκτός συνδρομής. Μια από τις δυνατότητες που μας προσφέρει η συνδρομή στο Proxmox, είναι η πρόσβαση σε πακέτα αναβαθμίσεων που έχουν περάσει από εκτεταμένο έλεγχο πριν την δημοσιοποίησή τους. Έτσι συνίσταται σε περιβάλλοντα παραγωγής εταιριών να έχουν ενεργή συνδρομή ώστε να μειώνετε η πιθανότητα προβλημάτων κατά την αναβάθμιση. Μπορούμε όμως να απενεργοποιήσουμε τα πακέτα αναβαθμίσεων μιας και δεν έχουμε πρόσβαση σε αυτά χωρίς ενεργή συνδρομή. Από το αριστερό μενού του διαχειριστικού περιβάλλοντος του Proxmox, επιλέγουμε από την λίστα την 2^η επιλογή στην σειρά όπου είναι και ο υπολογιστής που έχουμε εγκαταστήσει το Proxmox. Στην συνέχεια από την στήλη στα δεξιά του επεκτείνουμε την λίστα «Updates» και στην συνέχεια επιλέγουμε «Repositories» όπως φαίνεται και στην Εικόνα 19.



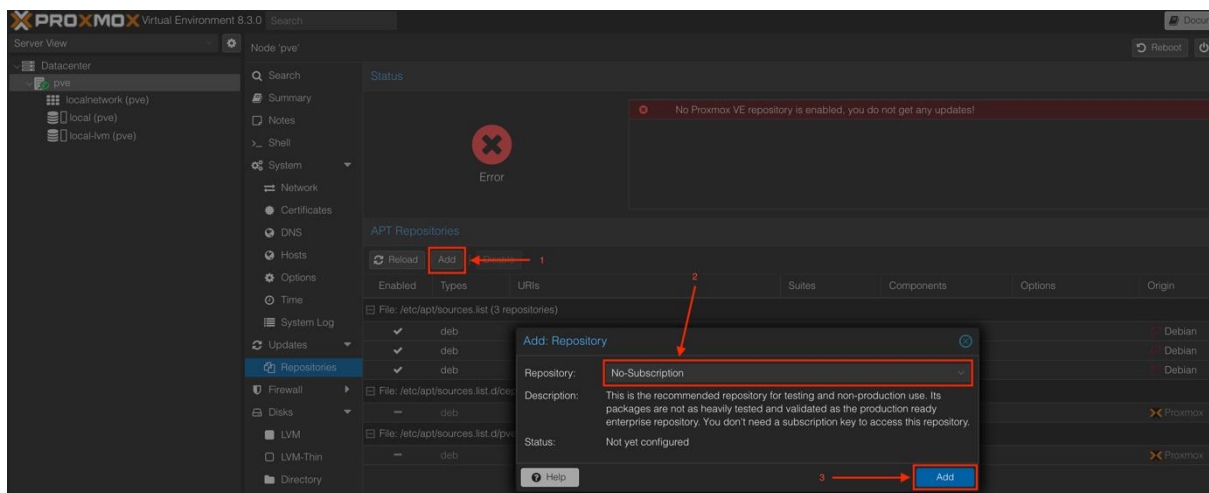
Εικόνα 19 Διαχείριση των ενημερώσεων του Proxmox.

Από την λίστα των «APT Repositories» επιλέγουμε όλες τις εγγραφές που ξεκινούν με «https://enterprise.proxmox.com» στο URI τους και στην συνέχεια πατάμε το κουμπί «Disable» ώστε να τις απενεργοποιήσουμε. Στην δική μας περίπτωση υπήρχαν 2 τέτοιες εγγραφές που τις απενεργοποιήσαμε όπως φαίνεται στην Εικόνα 20.



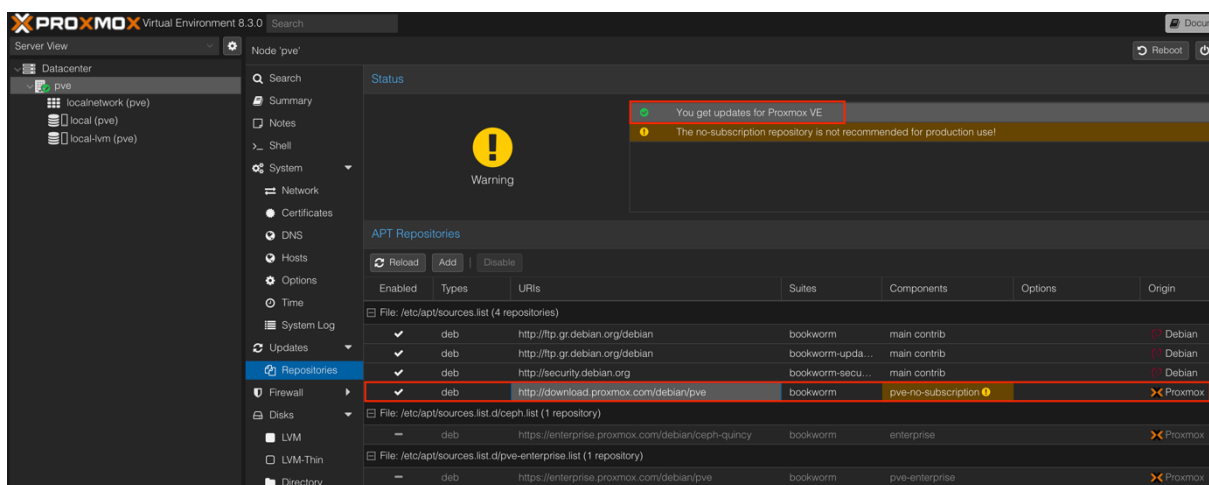
Εικόνα 20 Απενεργοποίηση των enterprise APT repositories στο Proxmox.

Τέλος, στην ίδια οθόνη πατάμε το κουμπί «Add», αφού επεκτείνουμε την λίστα που εμφανίζεται στο νέο παράθυρο, επιλέγουμε την επιλογή «No-Subscription» και πατάμε το κουμπί «Add» όπως φαίνεται στην Εικόνα 21.



Εικόνα 21 Ενεργοποίηση του No-Subscription APT repository στο Proxmox.

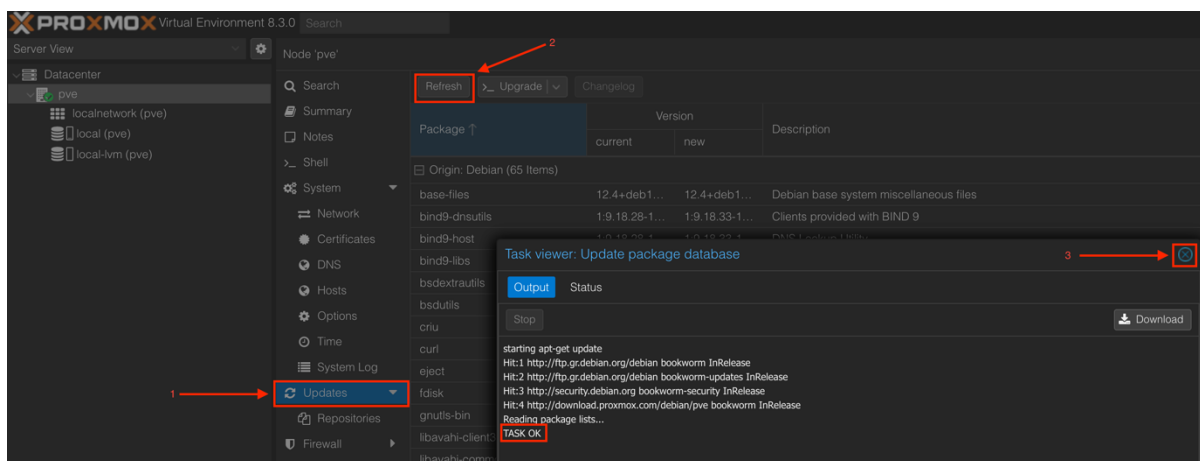
Μπορούμε να επιβεβαιώσουμε ότι η διαδικασία ολοκληρώθηκε επιτυχώς αν δούμε στον πίνακα «Status» στην ίδια οθόνη την ένδειξη «You get updates for Proxmox VE» όπως φαίνεται στην Εικόνα 22.



Εικόνα 22 Επιτυχής ενεργοποίηση των ενημερώσεων του Proxmox χωρίς ενεργή συνδρομή.

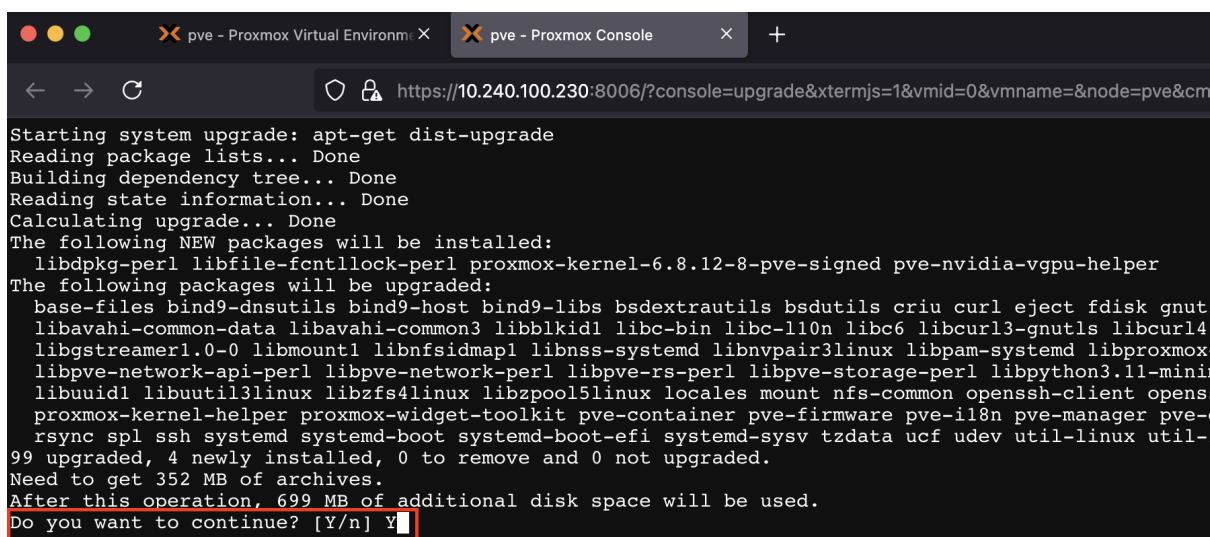
4.1.6 Αναβάθμιση του Proxmox στην τελευταία έκδοση

Οι αναβάθμιση του Proxmox είναι μια πολύ σημαντική εργασία που πρέπει να γίνεται τακτικά, για λόγους ασφάλειας και σταθερότητας του συστήματος μας. Η διαδικασία που πρέπει να ακολουθούμε είναι αρκετά απλή και μπορεί να γίνει εξολοκλήρου μέσα από το διαχειριστικό περιβάλλον του Proxmox. Αρχικά από το αριστερό μενού του διαχειριστικού περιβάλλοντος του Proxmox, επιλέγουμε από την λίστα την 2^η επιλογή στην σειρά όπου είναι και ο υπολογιστής που έχουμε εγκαταστήσει το Proxmox. Στην συνέχεια από την στήλη στα δεξιά του επιλέγουμε «Updates», στην συνέχεια πατάμε το κουμπί «Refresh» και περιμένουμε να ολοκληρωθεί η διαδικασία. Όταν μας επιστρέψει «TASK OK» όπως φαίνεται στην Εικόνα 23, μπορούμε να κλείσουμε το παράθυρο που άνοιξε και να συνεχίσουμε στο επόμενο βήμα.



Εικόνα 23 Πρώτο βήμα ενημέρωσης του Proxmox στην τελευταία έκδοση.

Έπειτα πατάμε το κουμπί «Upgrade» το οποίο θα μας ανοίξει ένα νέο παράθυρο στον περιηγητή μας και θα μας ζητείται να επιβεβαιώσουμε την διαδικασία πατώντας στο πληκτρολόγιο μας τον αγγλικό χαρακτήρα «Y» και στην συνέχεια πατώντας το πλήκτρο «Enter» όπως φαίνεται στην Εικόνα 24.



Εικόνα 24 Δεύτερο βήμα ενημέρωσης του Proxmox στην τελευταία έκδοση.

Τέλος, περιμένουμε να ολοκληρωθεί η διαδικασία και μπορούμε να επιβεβαιώσουμε στην κατάσταση της από το τι μας επιστρέφει όταν τελειώσει. Όπως φαίνεται στην Εικόνα 25 το Proxmox έχει αναβαθμιστεί επιτυχώς στην τελευταία έκδοση και επίσης μας προτείνεται να κάνουμε μια επανεκκίνηση.

```

update-initramfs: Generating /boot/initrd.img-6.8.12-8-pve
Running hook script 'zz-proxmox-boot'..
Re-executing '/etc/kernel/postinst.d/zz-proxmox-boot' in new private mount namespace..
No /etc/kernel/proxmox-boot-uuids found, skipping ESP sync.
Couldn't find EFI system partition. It is recommended to mount it to /boot or /efi.
Alternatively, use --esp-path= to specify path to mount point.
Processing triggers for libc-bin (2.36-9+deb12u10) ...

Your System is up-to-date

Seems you installed a kernel update - Please consider rebooting
this node to activate the new kernel.

starting shell
root@pve:/#

```

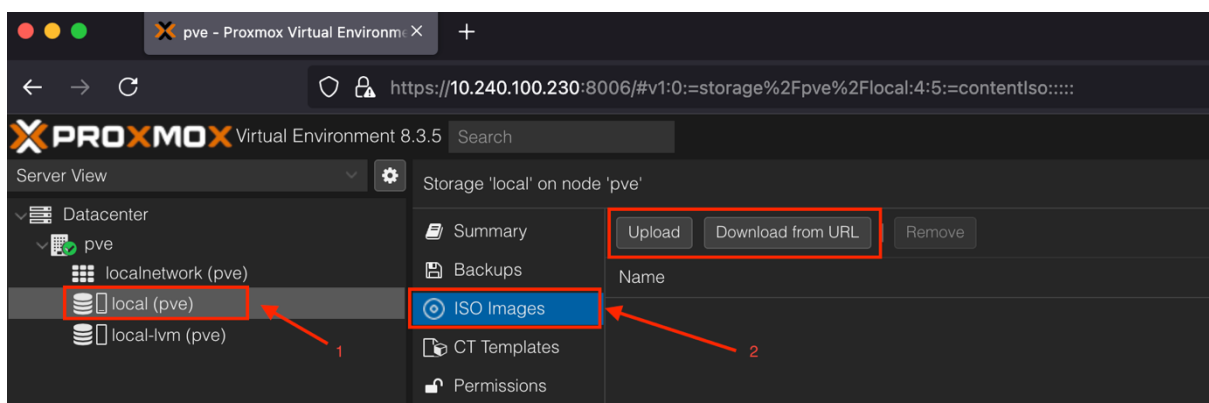
Εικόνα 25 Επιτυχής ενημέρωση του Proxmox στην τελευταία έκδοση.

4.2 Δημιουργία και Ρύθμιση των Εικονικών Μηχανών και Κοντέινερ

Σε αυτό το υποκεφάλαιο θα δημιουργήσουμε τις απαραίτητες εικονικές μηχανές και κοντέινερ, τα οποία θα χρησιμοποιήσουμε παρακάτω για την εγκατάσταση του Docker καθώς και την εγκατάσταση του Kubernetes.

4.2.1 Διαδικασία εισαγωγής ειδώλου εγκατάστασης (ISO) στο Proxmox

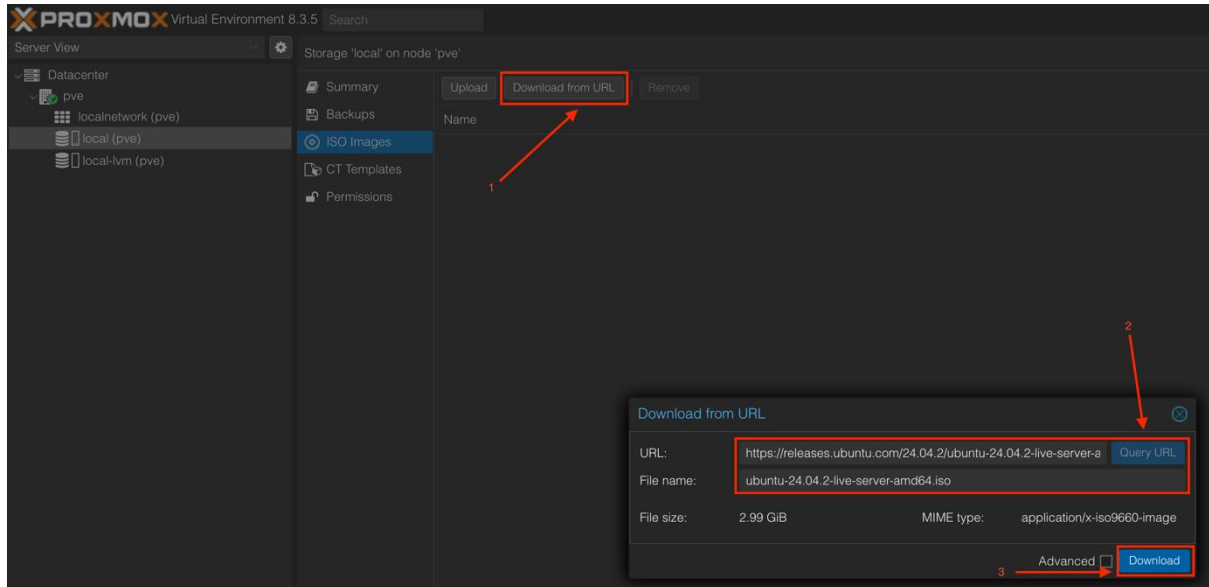
Για να δημιουργήσουμε μια εικονική μηχανή θα πρέπει να εισάγουμε στο Proxmox το είδωλο εγκατάστασης (ISO) του λειτουργικού συστήματος που θέλουμε να εγκαταστήσουμε. Στην περίπτωση μας επιλέγαμε το λειτουργικό σύστημα «Ubuntu 24.04.2 LTS Server» το οποίο και θα δείξουμε την διαδικασία λήψης του. Μέσα από το διαχειριστικό περιβάλλον του Proxmox μας δίνονται δυο επιλογές για να το πέτυχουμε αυτό. Η πρώτη είναι να το ανεβάσουμε από τον υπολογιστή μας και η δεύτερη να το κατεβάσει το Proxmox απευθείας δίνοντας του τον σύνδεσμο. Εμείς επιλέξαμε την δεύτερη επιλογή και έτσι από την αριστερή στήλη του διαχειριστικού περιβάλλοντος του Proxmox επιλέξαμε την 4η επιλογή από την αρχή «local (pve)» όπου είναι και ο φάκελος που θα αποθηκεύσει το Proxmox το είδωλο εγκαταστάσεις και στην συνέχεια επιλέξαμε ISO Images από την μεσαία στήλη. Έπειτα μπορούμε να διακρίνουμε τα 2 κουμπιά «Upload» και «Download from URL» όπως φαίνεται στην Εικόνα 26 που το κάθε ένα αντιπροσωπεύει αντίστοιχα τις επιλογές που αναφέραμε παραπάνω.



Εικόνα 26 Διαδικασία εισαγωγής ενός ειδώλου εγκατάστασης (ISO) στο Proxmox.

Στην συνέχεια πατάμε το κουμπί «Download from URL» και εισάγουμε το σύνδεσμο λήψης του ειδώλου εγκατάστασης στο πεδίο «URL» (Στην περίπτωση μας «<https://releases.ubuntu.com/24.04.2/ubuntu-24.04.2-live-server-amd64.iso>» το οποίο το

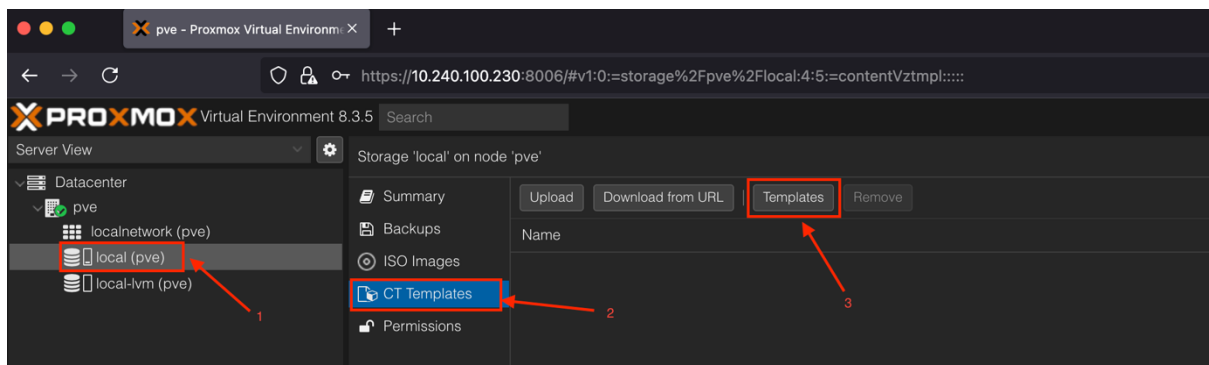
προμηθευτήκαμε από την επίσημη σελίδα της Canonical) και πατάμε το κουμπί «Query URL». Αφού μας εμφανίσει το όνομα του ειδώλου εγκατάστασης που θέλουμε να κάνουμε λήψη, πατάμε το κουμπί «Download» όπως φαίνεται στην Εικόνα 27 και περιμένουμε να ολοκληρωθεί η διαδικασία εμφανίζοντας «TASK OK». Έπειτα μπορούμε να κλείσουμε το παράθυρο που έχει εμφανιστεί.



Εικόνα 27 Λήψη του ειδώλου εγκατάστασης (ISO) στο Proxmox.

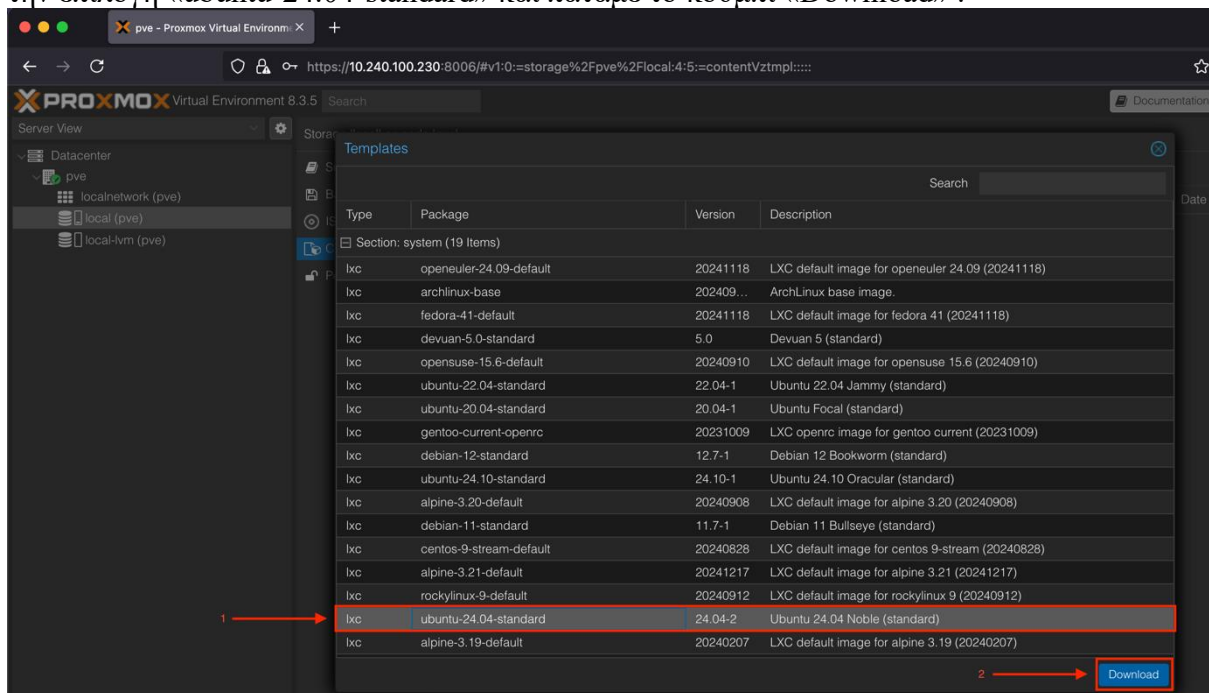
4.2.2 Διαδικασία εισαγωγής προτύπου LXC κοντέινερ στο Proxmox

Ομοίως με τις εικονικές μηχανές, για να δημιουργήσουμε ένα LXC κοντέινερ θα πρέπει να εισάγουμε στο Proxmox το πρότυπο που επιθυμούμε. Στην περίπτωση μας επιλέγαμε το λειτουργικό σύστημα «Ubuntu 24.04.2 LTS Server» για να υπάρχει ομοιομορφία με τις εικονικές μηχανές. Μέσα από το διαχειριστικό περιβάλλον του Proxmox μας δίνονται τρεις επιλογές για να εισάγουμε ένα πρότυπο LXC κοντέινερ. Οι δυο πρώτες είναι ίδιες με αυτές που ισχύουν για τα είδωλα εγκατάστασης (ISO) που εξηγήσαμε αναλυτικά στο υποκεφάλαιο 4.2.1. Η τρίτη επιλογή, που θα χρησιμοποιηθεί εδώ, είναι να κατεβάσουμε το πρότυπο από μια προκαθορισμένη λίστα που μας προσφέρει το διαχειριστικό περιβάλλον του Proxmox. Ξεκινώντας από την αριστερή στήλη του διαχειριστικού περιβάλλοντος του Proxmox επιλέξαμε την 4η επιλογή από την αρχή «local (pve)» όπου είναι και ο φάκελος που θα αποθηκεύσει το Proxmox το πρότυπο LXC κοντέινερ και στην συνέχεια επιλέξαμε «CT Templates» από την μεσαία στήλη. Έπειτα πατάμε το κουμπί «Templates» όπως φαίνεται στην Εικόνα 28.



Εικόνα 28 Διαδικασία εισαγωγής ενός προτύπου LXC κοντέινερ στο Proxmox.

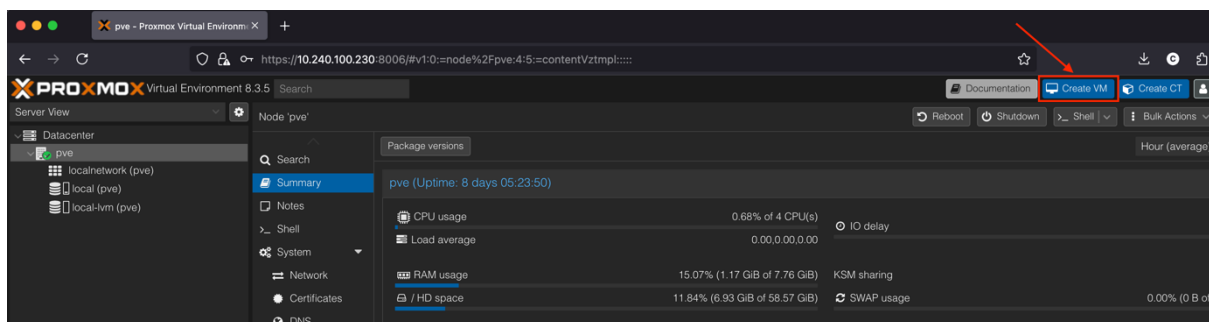
Στην συνέχεια, όπως φαίνεται στην Εικόνα 29, διαλέγουμε από την λίστα που εμφανίζεται την επιλογή «ubuntu-24.04-standard» και πατάμε το κουμπί «Download» .



Εικόνα 29 Λήψη ενός προτύπου LXC κοντέινερ μέσα από την λίστα του Proxmox.

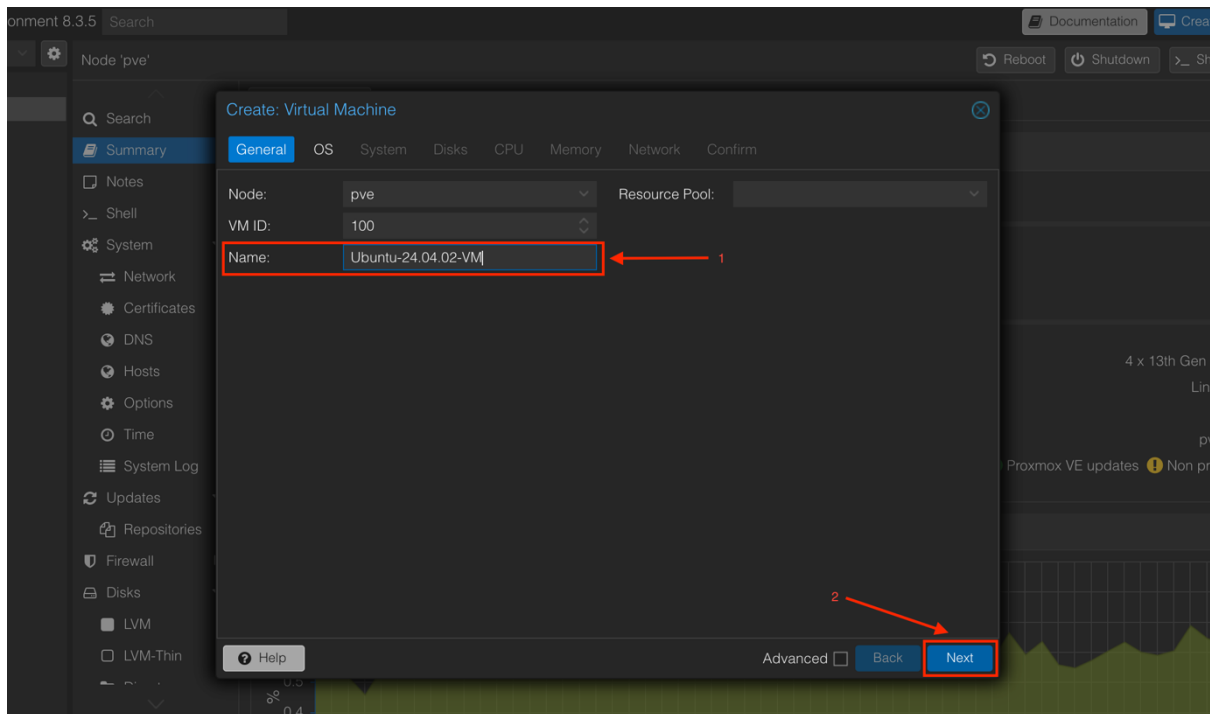
4.2.3 Διαδικασία δημιουργίας εικονικής μηχανής στο Proxmox

Η διαδικασία δημιουργίας μιας εικονικής μηχανής στο Proxmox είναι μια σχετικά απλή διαδικασία. Ο οδηγός που προσφέρει το διαχειριστικό περιβάλλον του Proxmox, μας καθοδηγεί ανάλογα με το λειτουργικό σύστημα που θέλουμε να εγκαταστήσουμε και μας προτείνει τις βέλτιστες ρυθμίσεις για αυτό. Η διαδικασία χρησιμοποιώντας τον οδηγό χωρίζεται σε 8 στάδια, στην περίπτωση μας θα δημιουργήσουμε μια εικονική μηχανή με λειτουργικό σύστημα Ubuntu 24.04.2 LTS Server. Μπορούμε να ξεκινήσουμε την διαδικασία δημιουργίας μιας εικονικής μηχανής από οποιαδήποτε οθόνη στο διαχειριστικό περιβάλλον του Proxmox βρισκόμαστε, πατώντας στην πάνω δεξιά γωνιά το κουμπί «Create VM» το οποίο θα μας εμφανίσει τον οδηγό όπως φαίνεται στην Εικόνα 30.



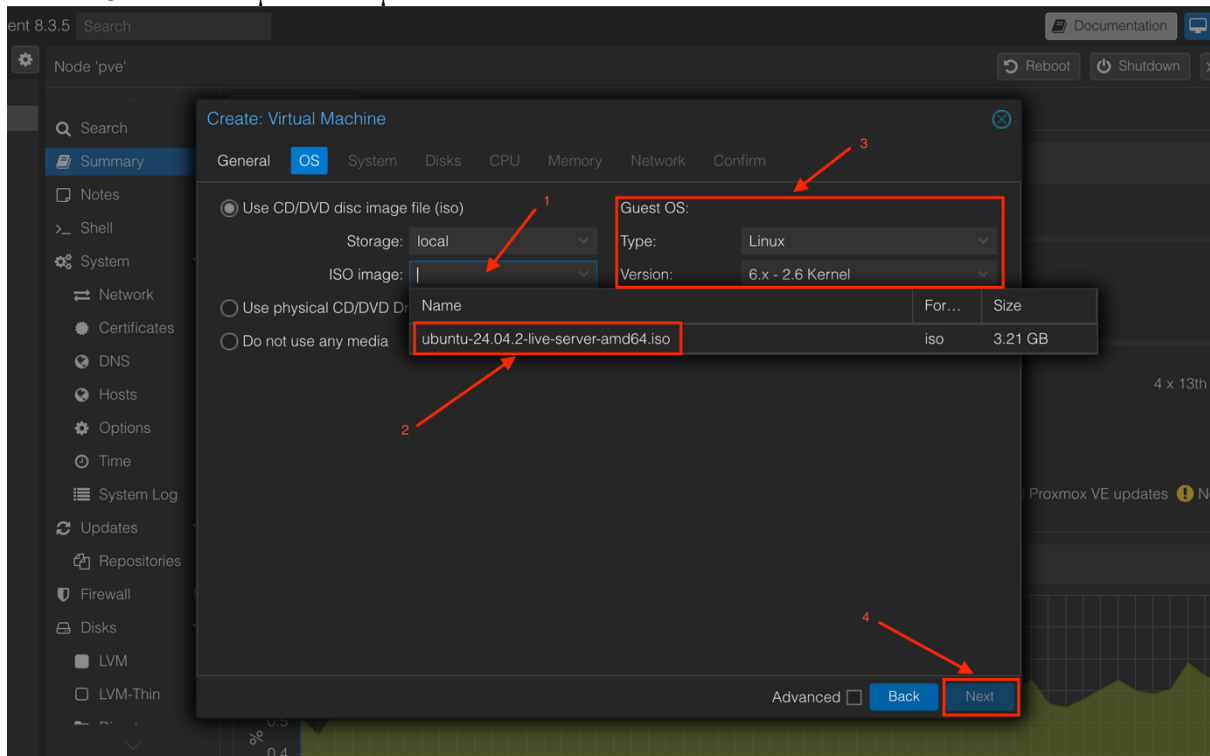
Εικόνα 30 Κουμπί δημιουργίας εικονικής μηχανής στο διαχειριστικό περιβάλλον του Proxmox.

Στο πρώτο στάδιο «General», στο πεδίο «Name» εισάγουμε το όνομα της εικονικής μηχανής και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 31.



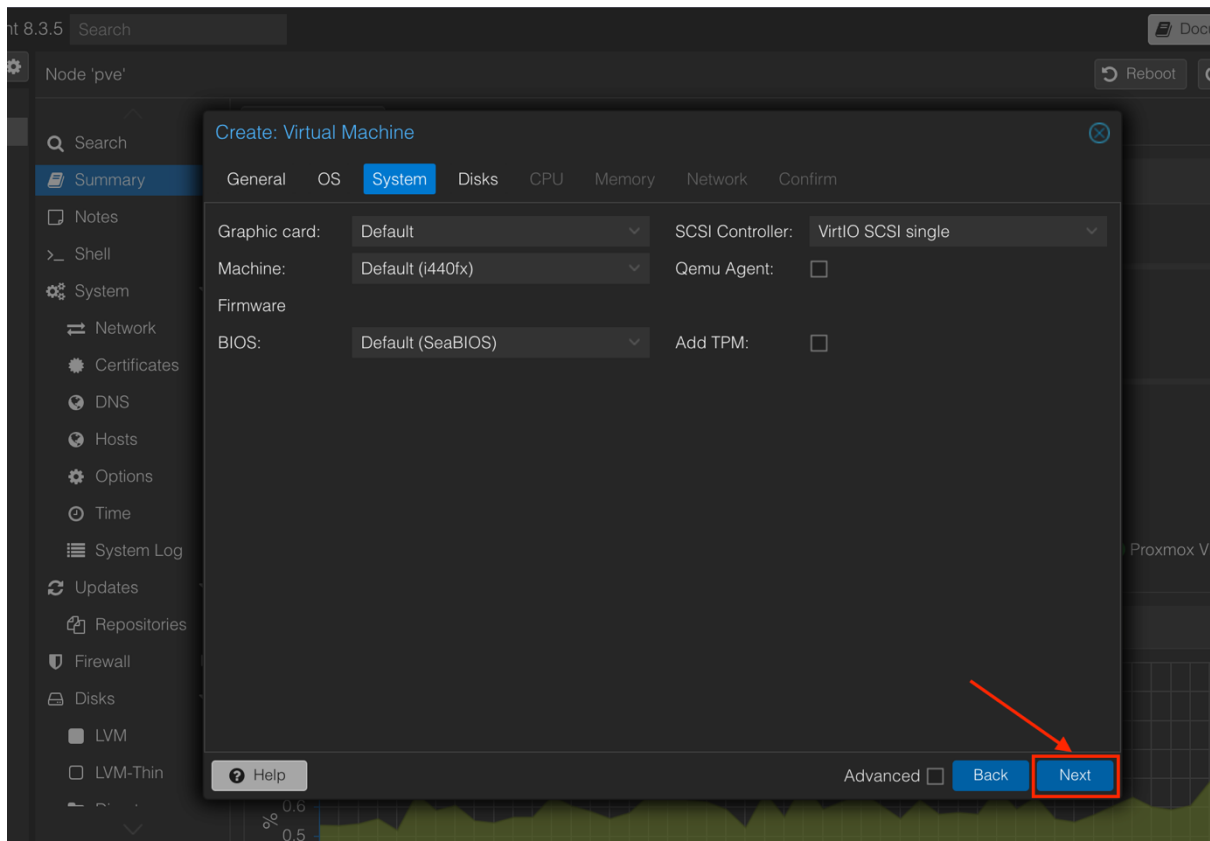
Εικόνα 31 Πρώτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.

Στο δεύτερο στάδιο «OS», επεκτείνουμε την λίστα «ISO Image» και επιλέγουμε το είδωλο εγκατάστασης (ISO) που επιθυμούμε και στην συνέχεια στις δυο λίστες στα δεξιά επιλέγουμε τις επιθυμητές ρυθμίσεις για το λειτουργικό που θέλουμε να εγκαταστήσουμε. Στην περίπτωση μας επιλέγουμε τύπος «Linux» και έκδοση «6.x - 2.6 Kernel» όπως φαίνεται στην Εικόνα 32 και πατάμε το κουμπί «Next».



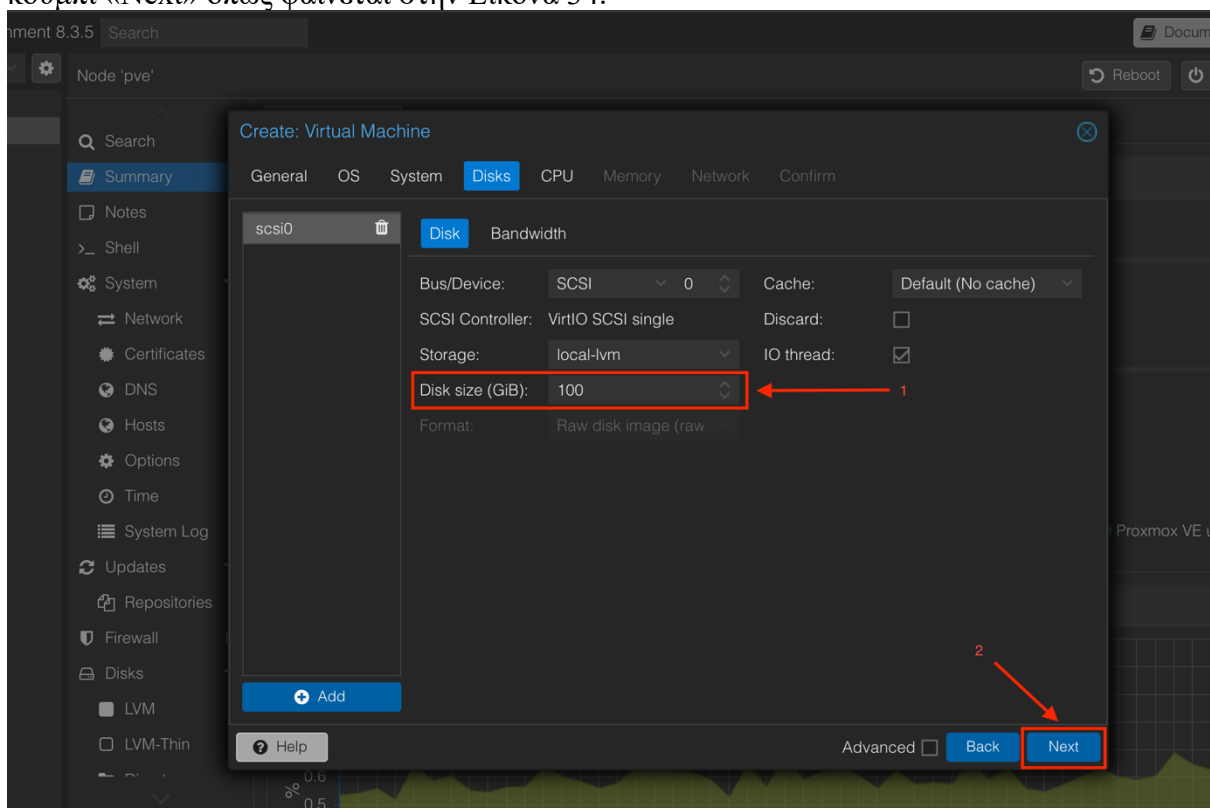
Εικόνα 32 Δεύτερο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.

Στο τρίτο στάδιο «System», αφήνουμε τις προεπιλεγμένες ρυθμίσεις και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 33.



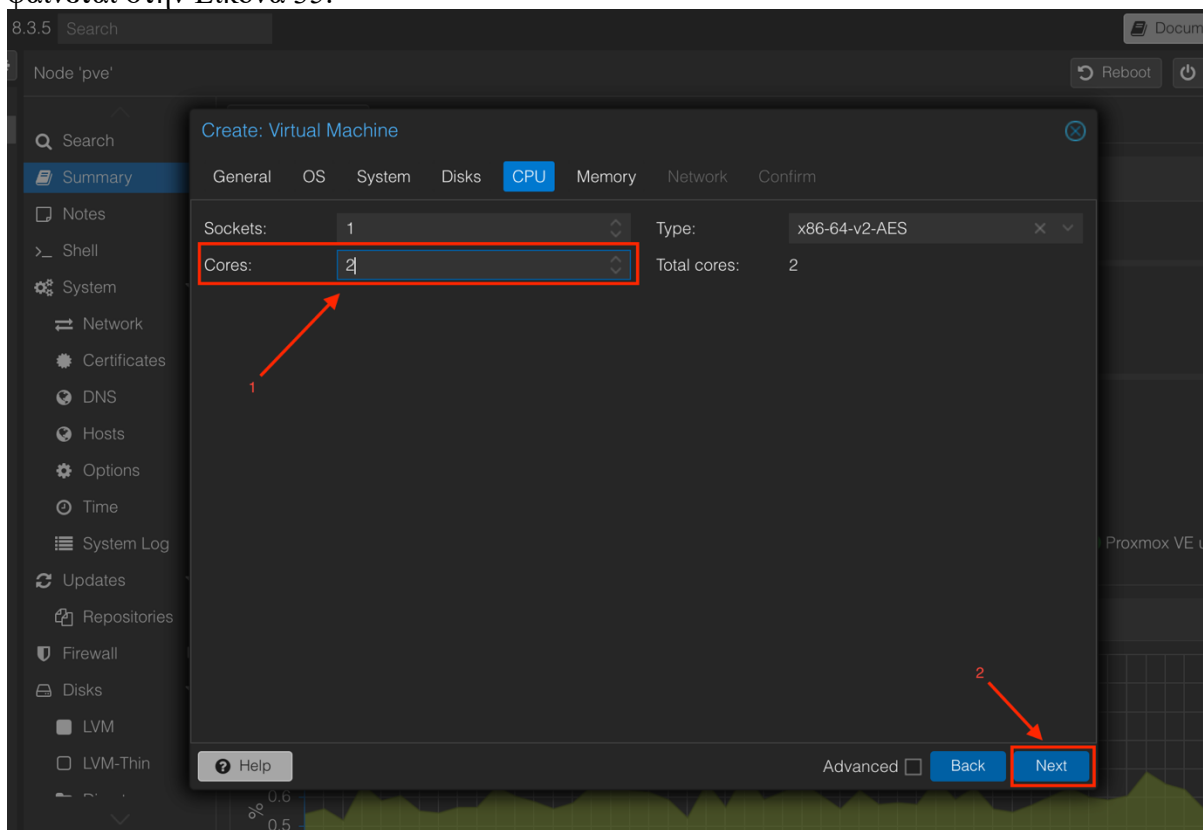
Εικόνα 33 Τρίτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.

Στο τέταρτο στάδιο «Disks», εισάγουμε στο πεδίο «Disk size (GiB)» την επιθυμητή χωρητικότητα του σκληρού δίσκου σε μονάδα μέτρησης GiB (Gibibyte) και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 34.



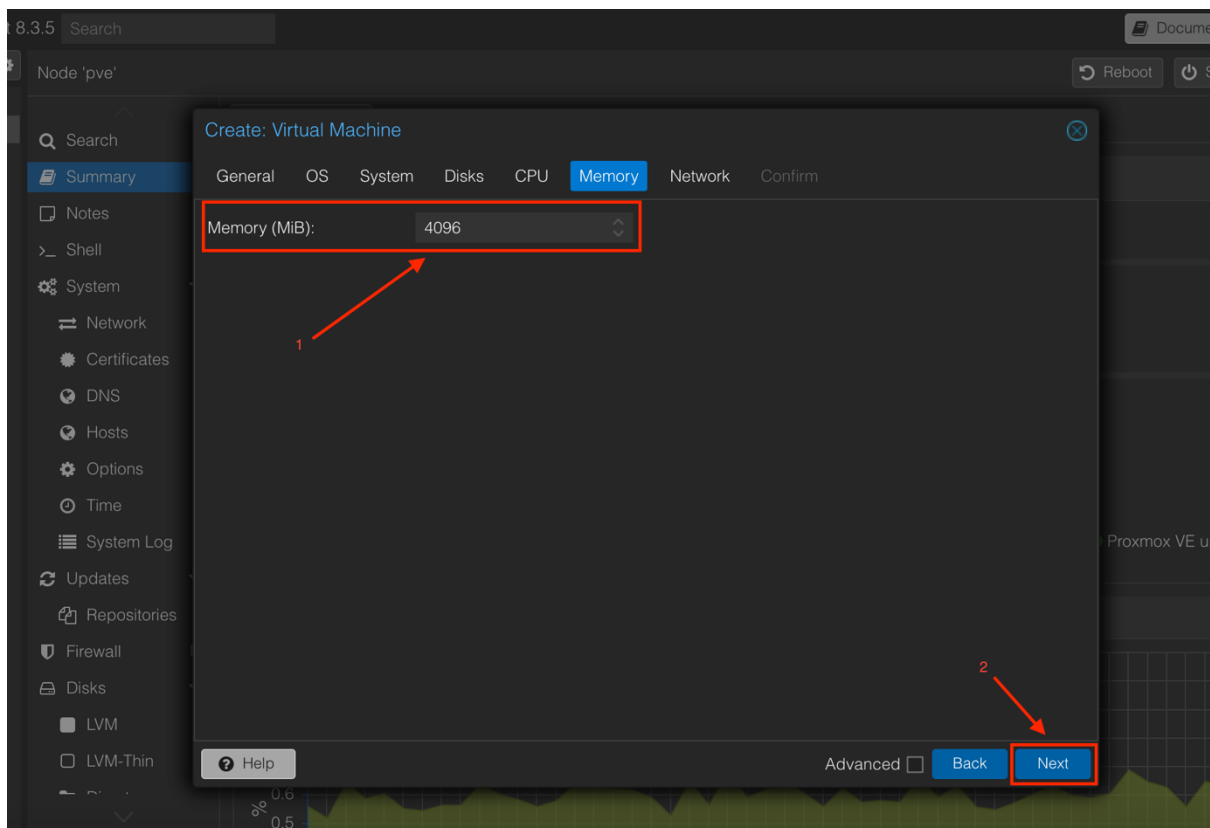
Εικόνα 34 Τέταρτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.

Στο πέμπτο στάδιο «CPU», εισάγουμε στο πεδίο «Cores» τον αριθμό των πυρήνων του επεξεργαστή που επιθυμούμε να έχει η εικονική μηχανή και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 35.



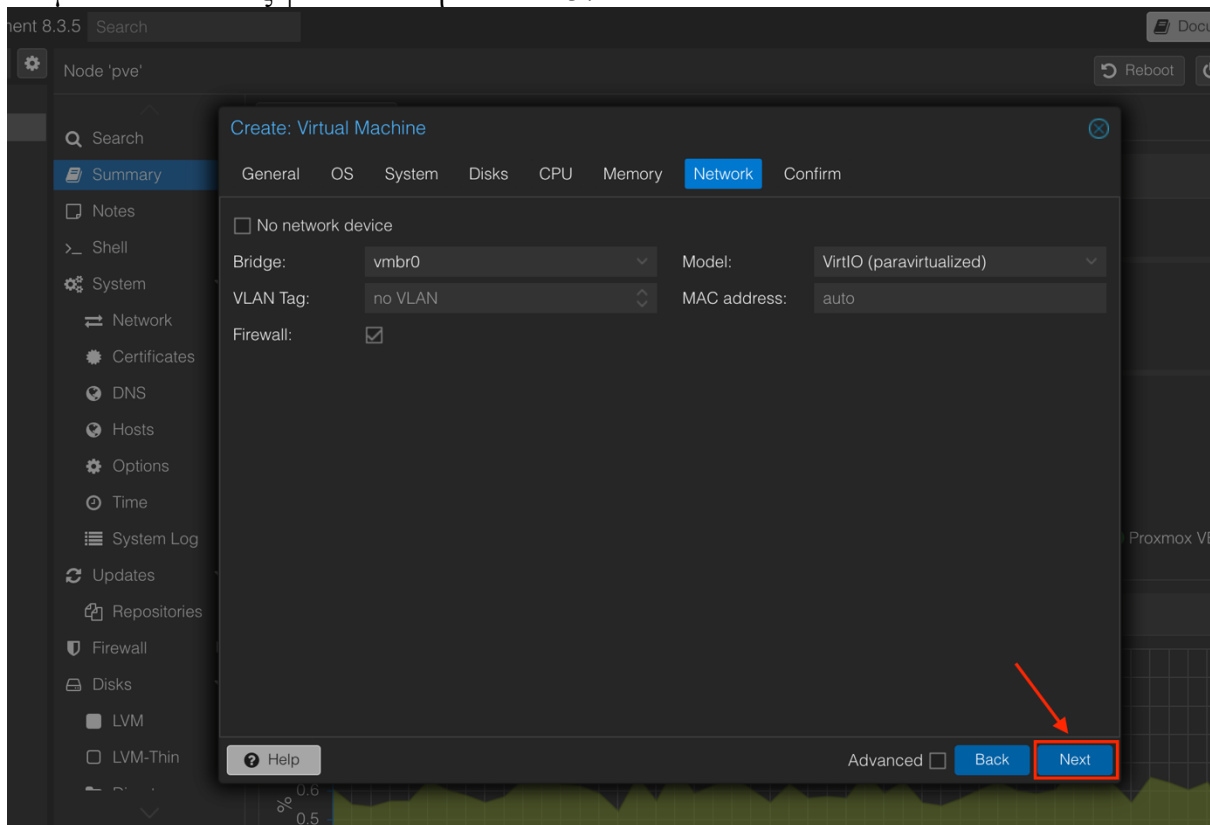
Εικόνα 35 Πέμπτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.

Στο έκτο στάδιο «Memory», εισάγουμε στο πεδίο «Memory (MiB)» το μέγεθος σε μονάδα μέτρησης MiB (Mebibyte) της μνήμης συστήματος (RAM) που επιθυμούμε και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 36.



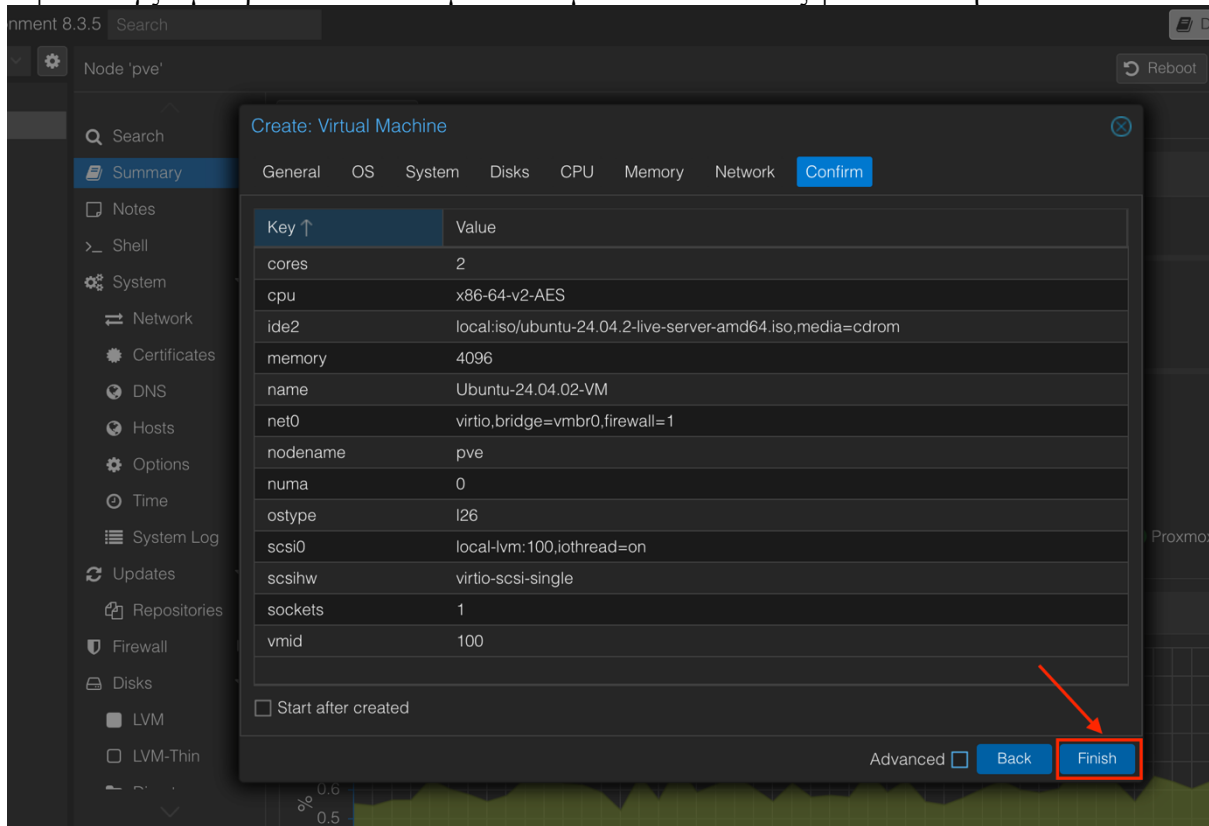
Εικόνα 36 Έκτο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.

Στο έβδομο στάδιο «Network», αφήνουμε τις προεπιλεγμένες ρυθμίσεις και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 37.



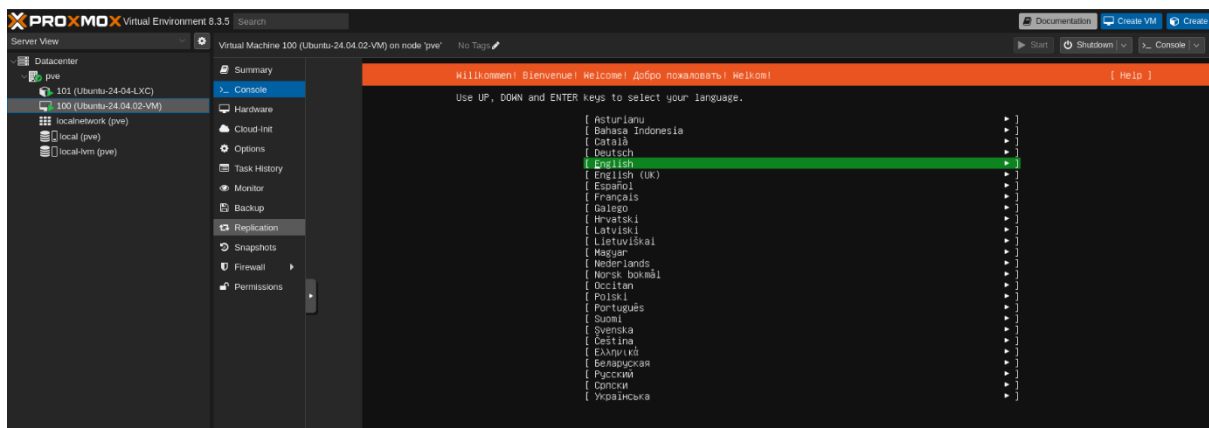
Εικόνα 37 Εβδομο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.

Στο όγδοο στάδιο «Confirm», μας προβάλλεται μια σύνοψη των επιλογών που έχουμε διαλέξει ώστε να κάνουμε έναν συνολικό έλεγχο πριν την δημιουργία της εικονικής μηχανής. Αφού ελέγξουμε προσεκτικά πατάμε το κουμπί «Finish» όπως φαίνεται στην Εικόνα 38.



Εικόνα 38 Όγδοο στάδιο του οδηγού δημιουργίας εικονικής μηχανής από το διαχειριστικό περιβάλλον του Proxmox.

Αμέσως μετά την ολοκλήρωση της δημιουργίας της εικονικής μηχανής, θα πρέπει να εγκαταστήσουμε το λειτουργικό σύστημα σε αυτή όπως θα κάναμε και σε ένα φυσικό μηχάνημα. Ξεκινήσαμε ανοίγοντας την κονσόλα της συγκεκριμένης εικονικής μηχανής και την εκκινήσαμε ακολουθώντας τις οδηγίες στο υποκεφάλαιο 4.2.5. Τέλος, ακολουθούμε τις οδηγίες που αναγράφονται στην επίσημη σελίδα της Canonical Ltd⁵, για να εγκαταστήσουμε τα Ubuntu 24.04.2 LTS Server όπως φαίνεται στην Εικόνα 39.

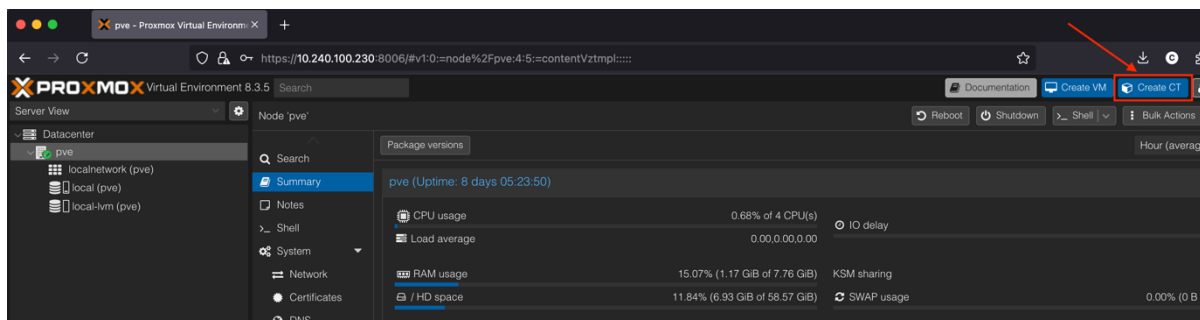


Εικόνα 39 Εγκατάσταση του λειτουργικού συστήματος Ubuntu 24.04.2 LTS Server στην εικονική μηχανή.

⁵ <https://ubuntu.com/tutorials/install-ubuntu-server#4-choose-your-language>

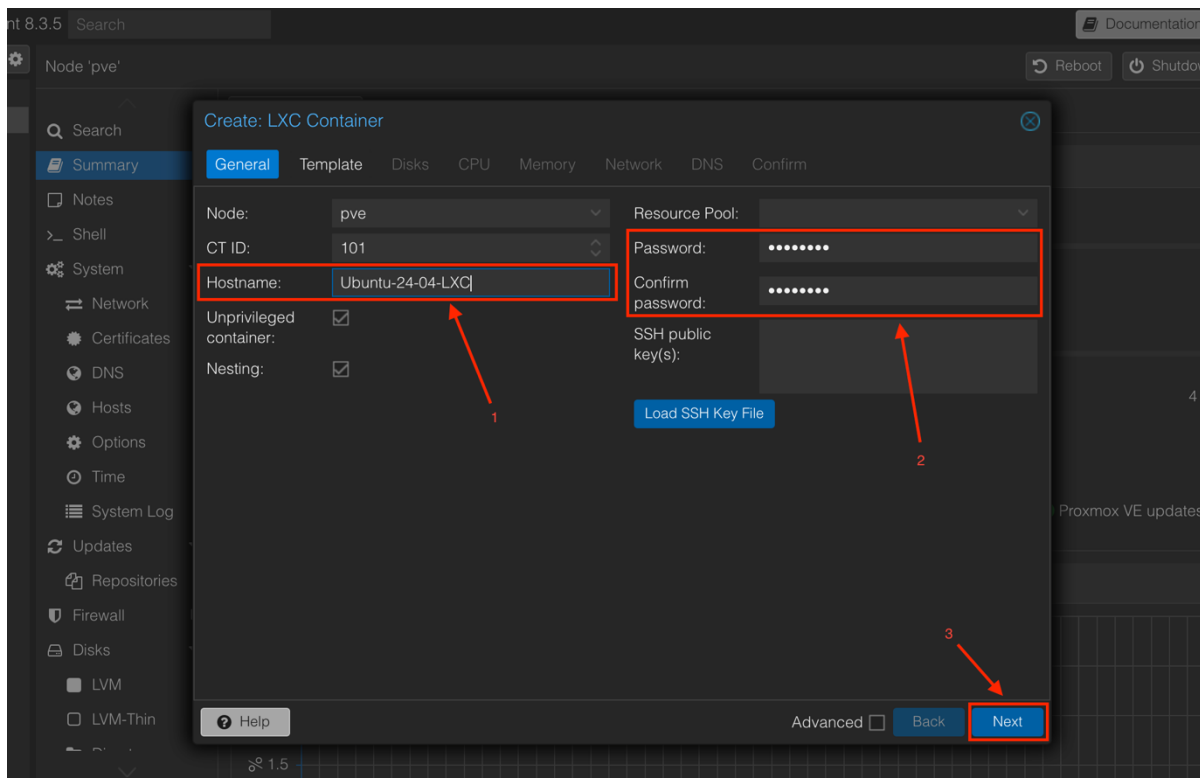
4.2.4 Διαδικασία δημιουργίας LXC κοντέινερ στο Proxmox

Η διαδικασία που αναφέραμε στο υποκεφάλαιο 4.2.3, για την δημιουργία μιας εικονικής μηχανής, έχει αρκετές ομοιότητες με την διαδικασία που θα ακολουθήσουμε για να δημιουργήσουμε ένα LXC κοντέινερ. Επιλέξαμε να δημιουργήσουμε ένα LXC κοντέινερ το οποίο έχει λειτουργικό «Ubuntu 24.04 LTS Server» ώστε να υπάρχει ομοιομορφία με την εικονική μηχανή που δημιουργήσαμε. Ξεκινάμε την διαδικασία δημιουργίας ενός LXC κοντέινερ, πατώντας το κουμπί «Create CT» στην πάνω δεξιά γωνιά από οποιαδήποτε οθόνη βρισκόμαστε στο διαχειριστικό περιβάλλον του Proxmox, το οποίο θα μας εμφανίσει τον οδηγό όπως φαίνεται στην Εικόνα 40.



Εικόνα 40 Κουμπί δημιουργίας LXC κοντέινερ στο διαχειριστικό περιβάλλον του Proxmox.

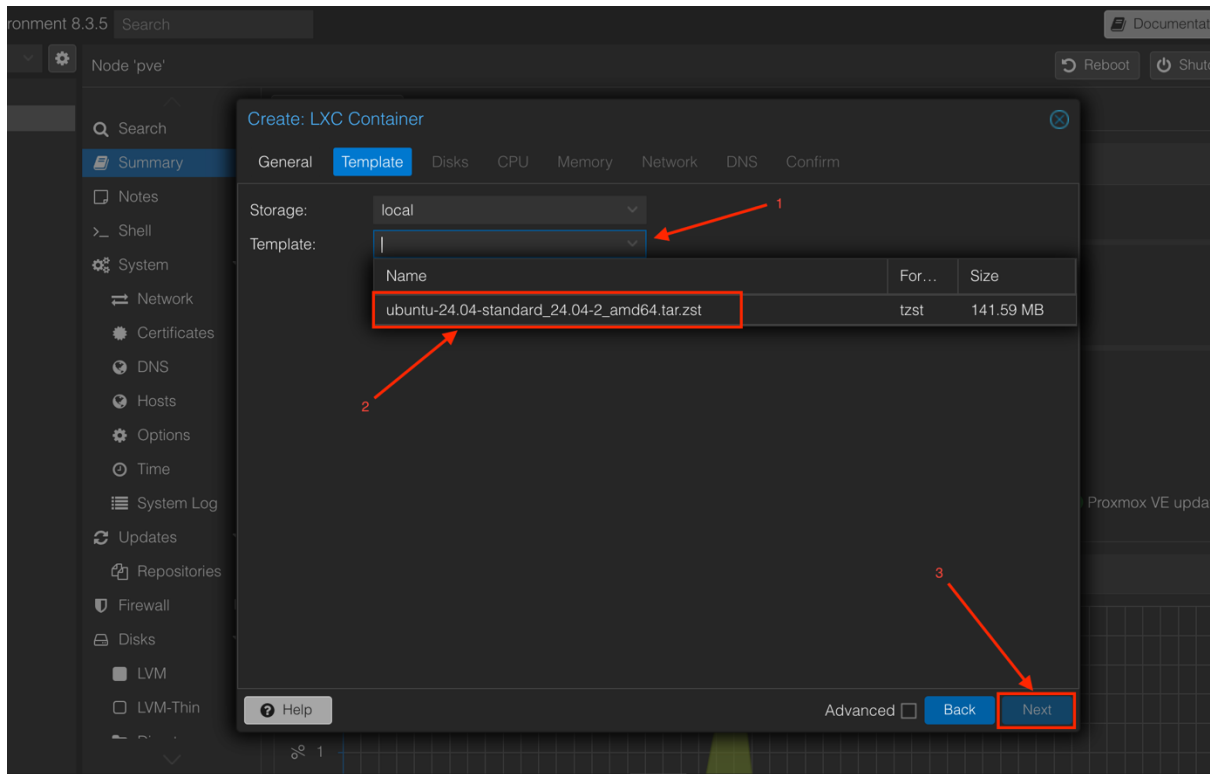
Στο πρώτο στάδιο «General», στο πεδίο «Hostname» εισάγουμε το όνομα του LXC κοντέινερ και έναν κωδικό πρόσβασης στα πεδία «Password» και «Confirm Password» με τον οποίο θα συνδεόμαστε μέσα στο τερματικό του LXC κοντέινερ. Αφού συμπληρώσουμε τα πεδία πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 41.



Εικόνα 41 Πρώτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.

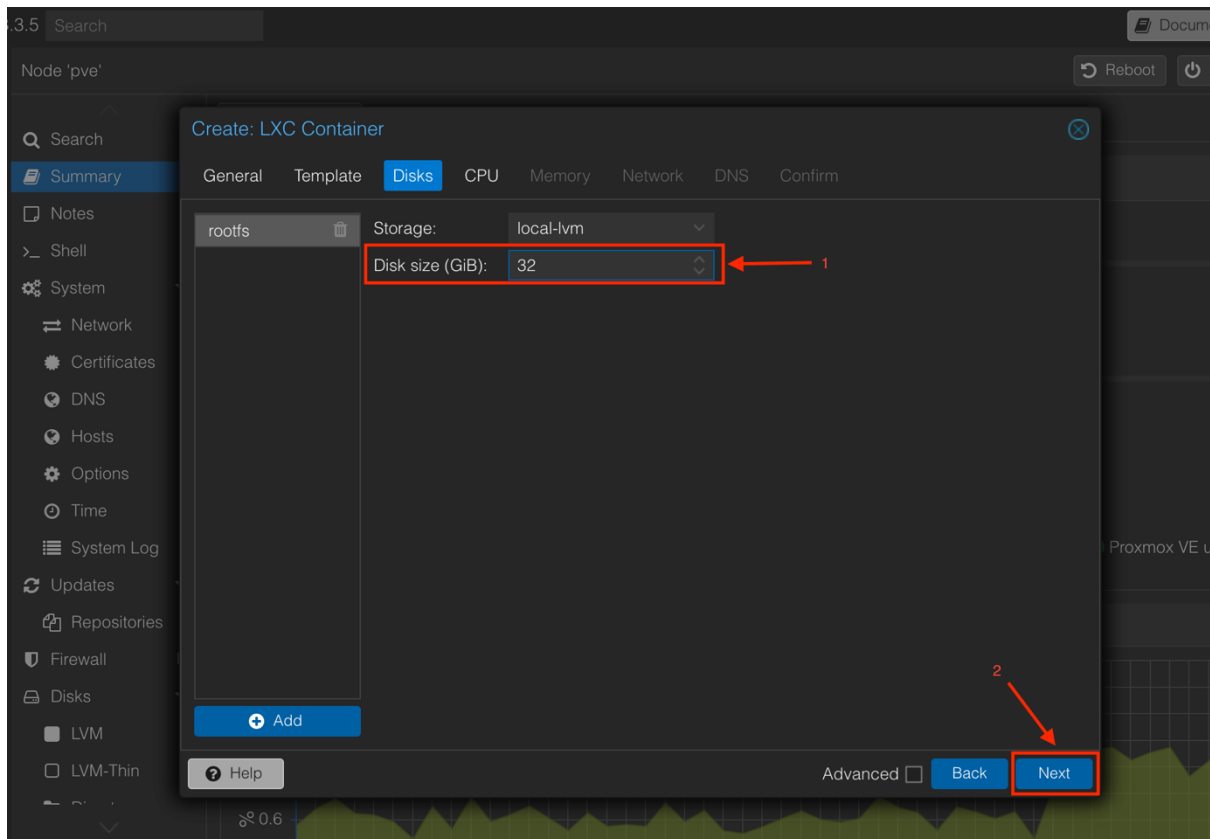
Στο δεύτερο στάδιο «Template», επεκτείνουμε την λίστα «Template» και επιλέγουμε το πρότυπο που επιθυμούμε να εγκαταστήσουμε στην δική μας περίπτωση «ubuntu-24.04-

standard_24.04-2_amd64.tar.zst» όπως φαίνεται στην Εικόνα 42 και πατάμε το κουμπί «Next».



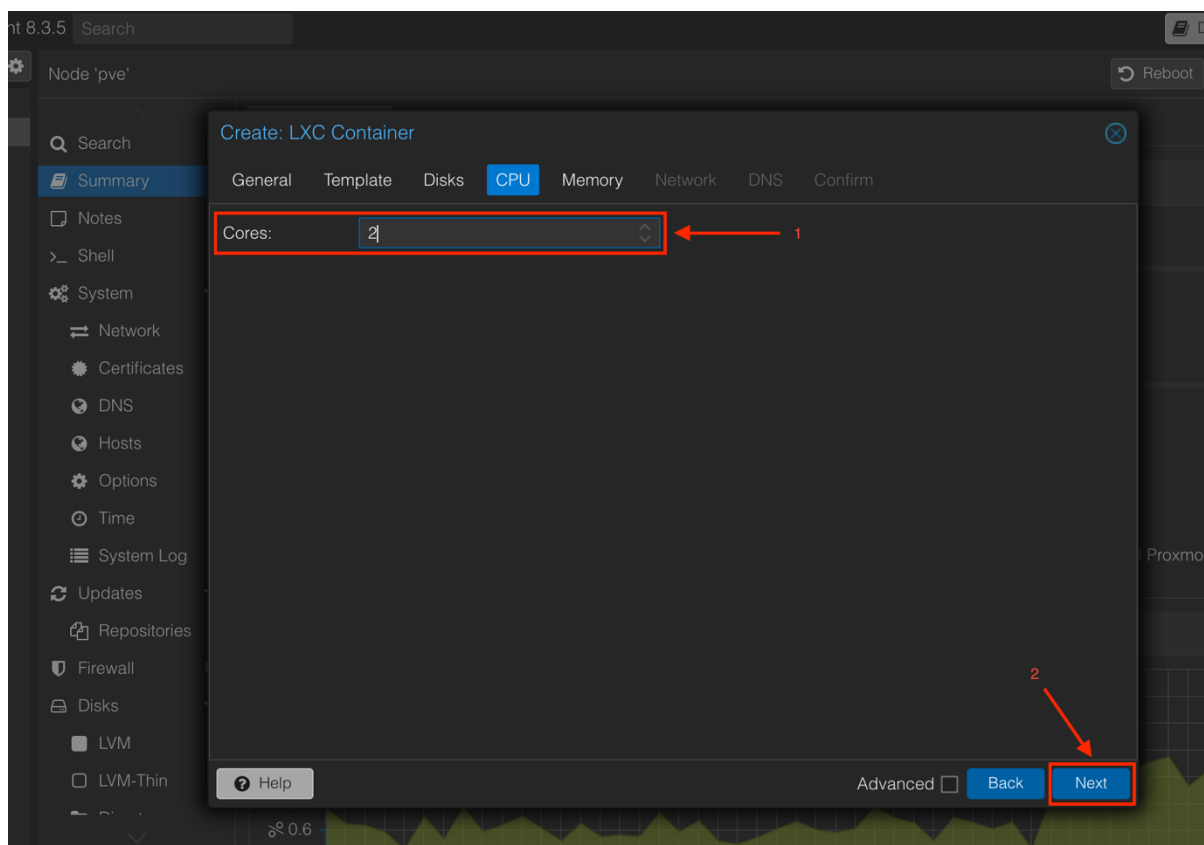
Εικόνα 42 Δεύτερο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.

Στο τρίτο στάδιο «Disks», εισάγουμε στο πεδίο «Disk size (GiB)» την επιθυμητή χωρητικότητα του σκληρού δίσκου σε μονάδα μέτρησης GiB (Gibibyte) και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 43.



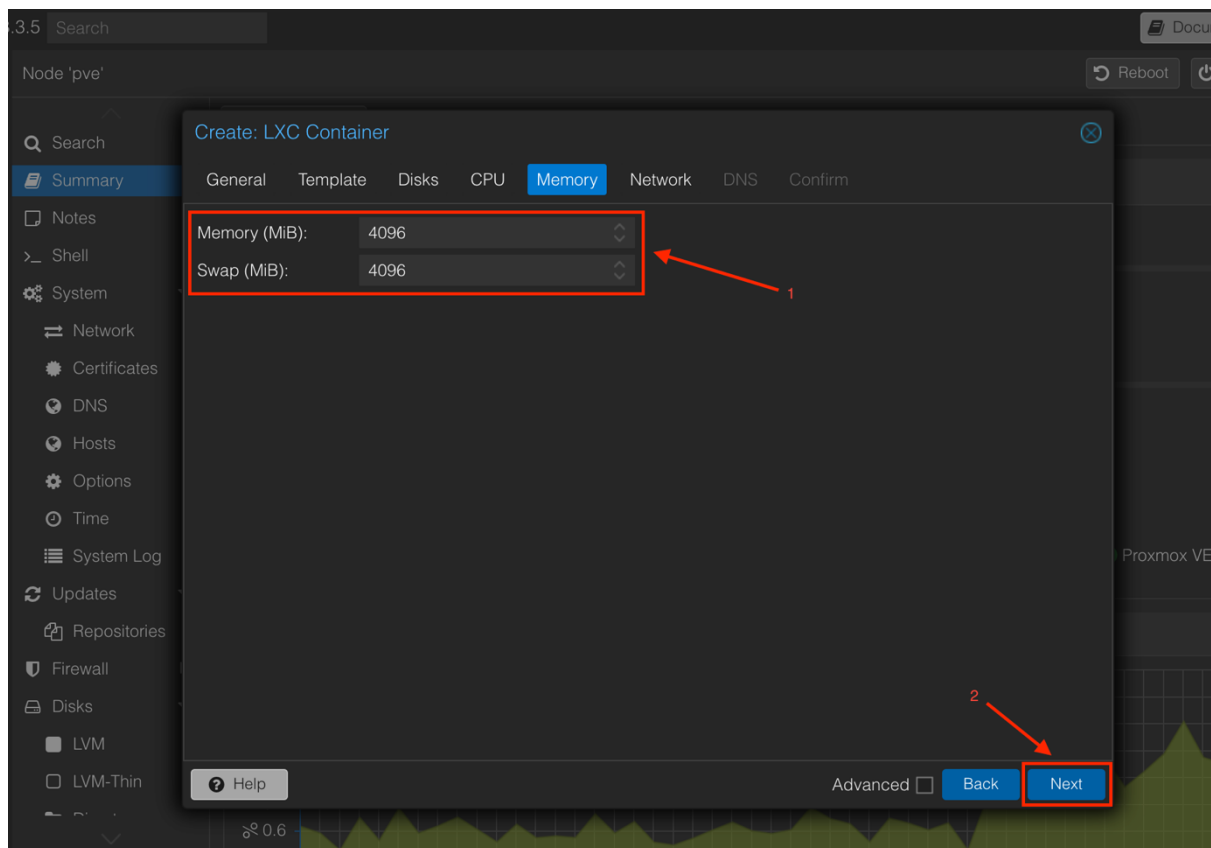
Εικόνα 43 Τρίτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.

Στο τέταρτο στάδιο «CPU», εισάγουμε στο πεδίο «Cores» τον αριθμό των πυρήνων του επεξεργαστή που επιθυμούμε να έχει το LXC κοντέινερ και στην συνέχεια πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 44.



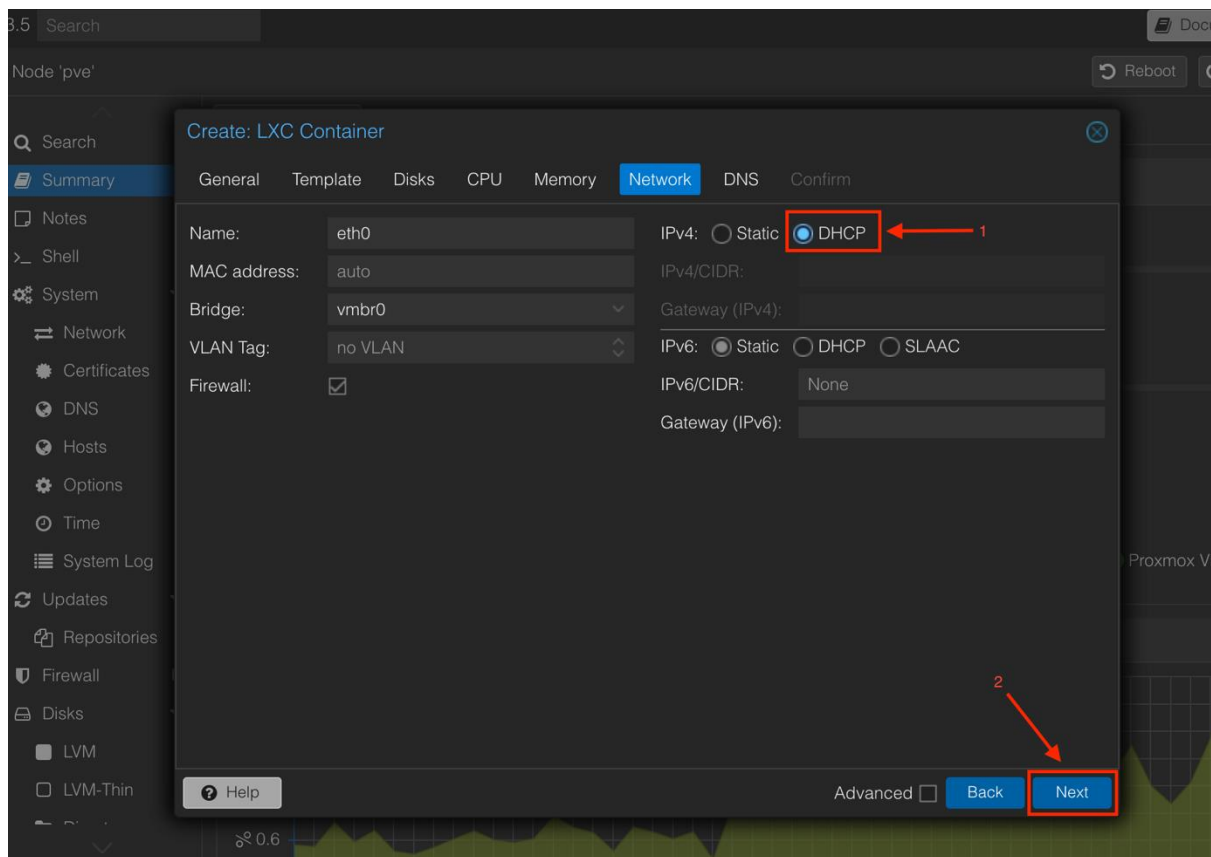
Εικόνα 44 Τέταρτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.

Στο πέμπτο στάδιο «Memory», εισάγουμε στα πεδία «Memory (MiB)» και «Swap (MiB)» το μέγεθος της μνήμης συστήματος (RAM) και της εικονικής μνήμης (SWAP) που επιθυμούμε αντίστοιχα, σε μονάδα μέτρησης MiB (Mebibyte) και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 45.



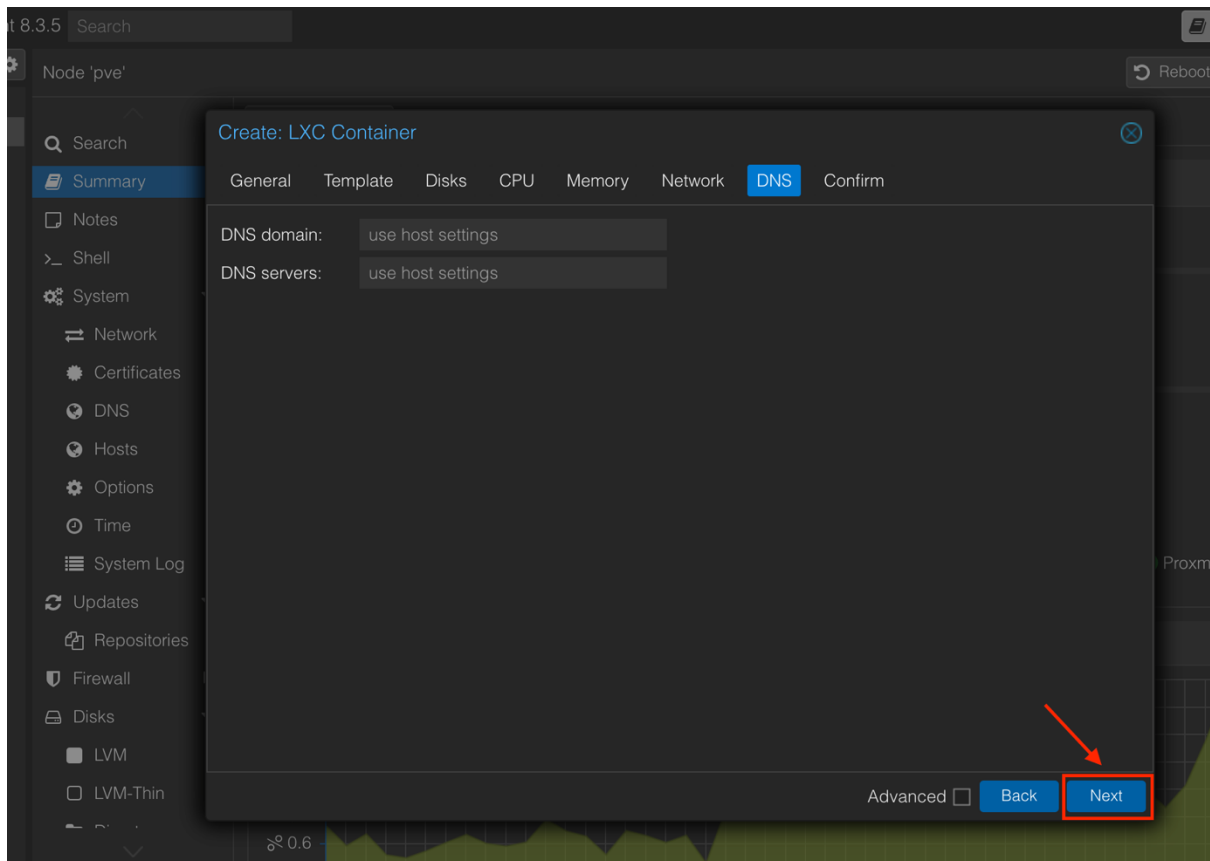
Εικόνα 45 Πέμπτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.

Στο έκτο στάδιο «Network», επιλέγουμε στο πεδίο «IPv4» στην επιλογή «DHCP» σύμφωνα με την μέθοδο διευθυνσιοδότησης του τοπικού μας δικτύου, και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 46.



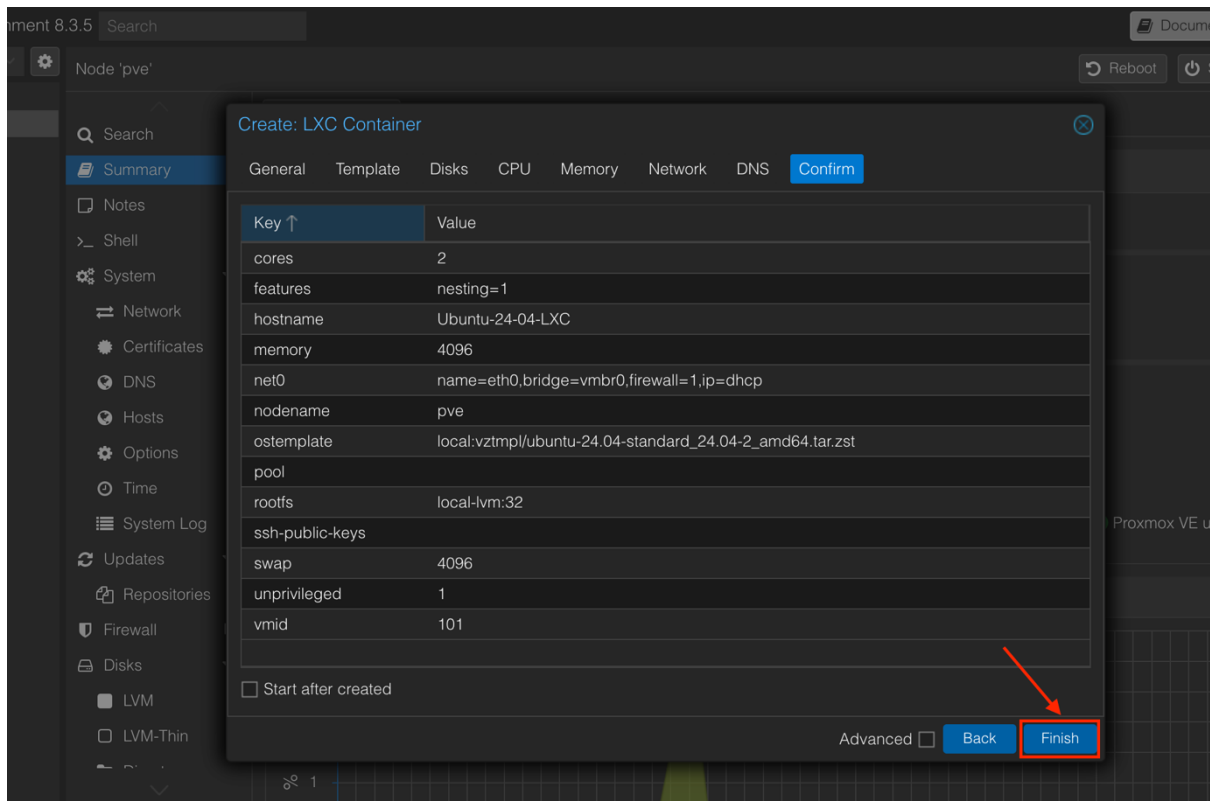
Εικόνα 46 Έκτο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.

Στο έβδομο στάδιο «DNS», αφήνουμε τις προεπιλεγμένες ρυθμίσεις και πατάμε το κουμπί «Next» όπως φαίνεται στην Εικόνα 47.



Εικόνα 47 Εβδομο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.

Τέλος, στο όγδοο στάδιο «Confirm», μας προβάλλεται μια σύνοψη των επιλογών που έχουμε διαλέξει ώστε να κάνουμε έναν συνολικό έλεγχο πριν την δημιουργία του LXC κοντέινερ. Αφού ελέγξουμε προσεκτικά πατάμε το κουμπί «Finish» όπως φαίνεται στην Εικόνα 48. Σε αυτό το σημείο δεν χρειάζεται να προχωρήσουμε στην εγκατάσταση του λειτουργικού συστήματος μέσα στο κοντέινερ όπως κάναμε για την εικονική μηχανή, διότι είναι ήδη προεγκατεστημένο στα κοντέινερ και προέρχεται μέσα από το πρότυπο κοντέινερ που επιλέξαμε.

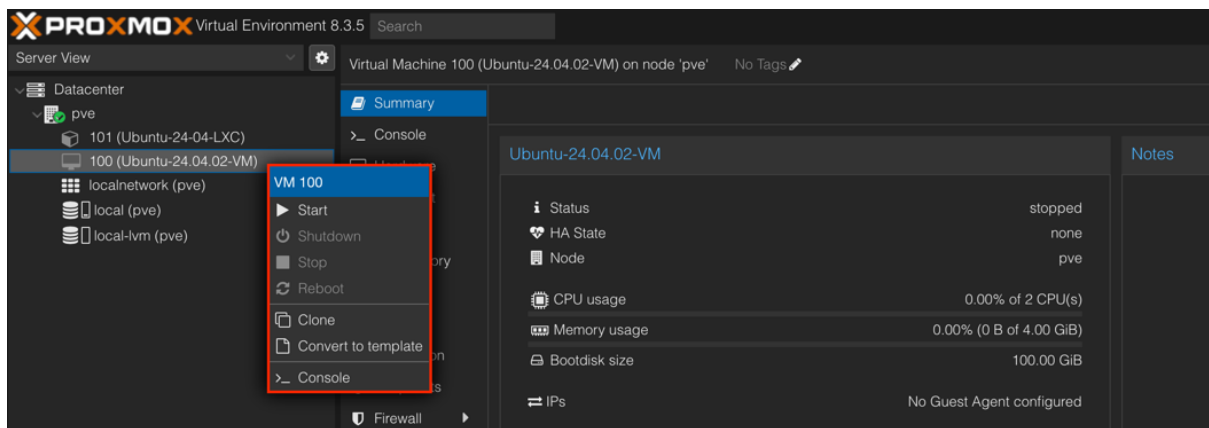


Εικόνα 48 Όγδοο στάδιο του οδηγού δημιουργίας LXC κοντέινερ από το διαχειριστικό περιβάλλον του Proxmox.

4.2.5 Διαχείριση εικονικών μηχανών και κοντέινερ μέσα από το Proxmox

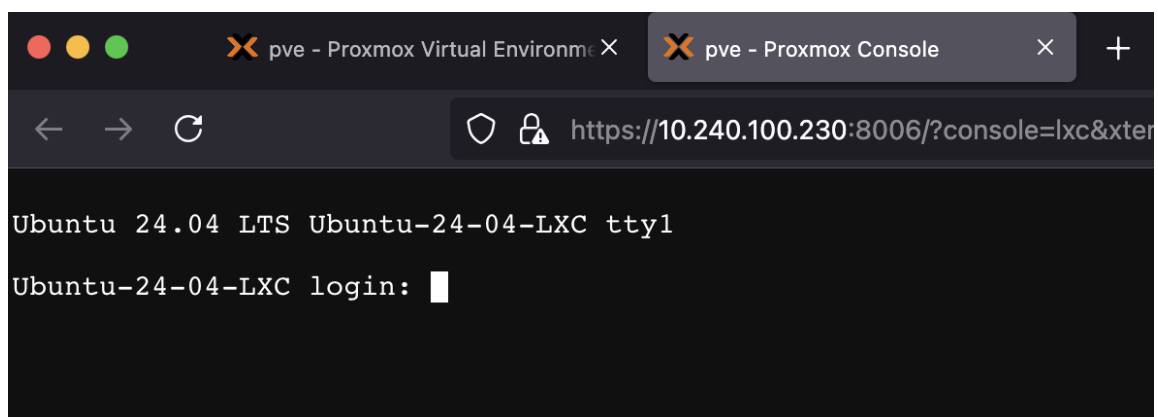
Το διαχειριστικό περιβάλλον του Proxmox μας επιτρέπει μέσα από αυτό να διαχειριζόμαστε το λειτουργικό που έχουμε εγκαταστήσει μέσα σε εικονικές μηχανές ή LXC κοντέινερ. Υπάρχουν αρκετοί τρόποι να επιτύχουμε το παραπάνω αποτέλεσμα. Ένας τρόπος είναι κάνοντας δεξί κλικ πάνω στο όνομα της εικονικής μηχανής ή του LXC κοντέινερ και μας εμφανίζεται ένα μενού όπως φαίνεται στην Εικόνα 49, το οποίο μας δίνει τις εξής επιλογές:

- Start – Εκκίνηση
- Shutdown – Τερματισμός
- Stop – Άμεσος τερματισμός
- Reboot – Επανεκκίνηση
- Clone – Κλωνοποίηση
- Convert to template – Μετατροπή σε πρότυπο
- Console – Κονσόλα



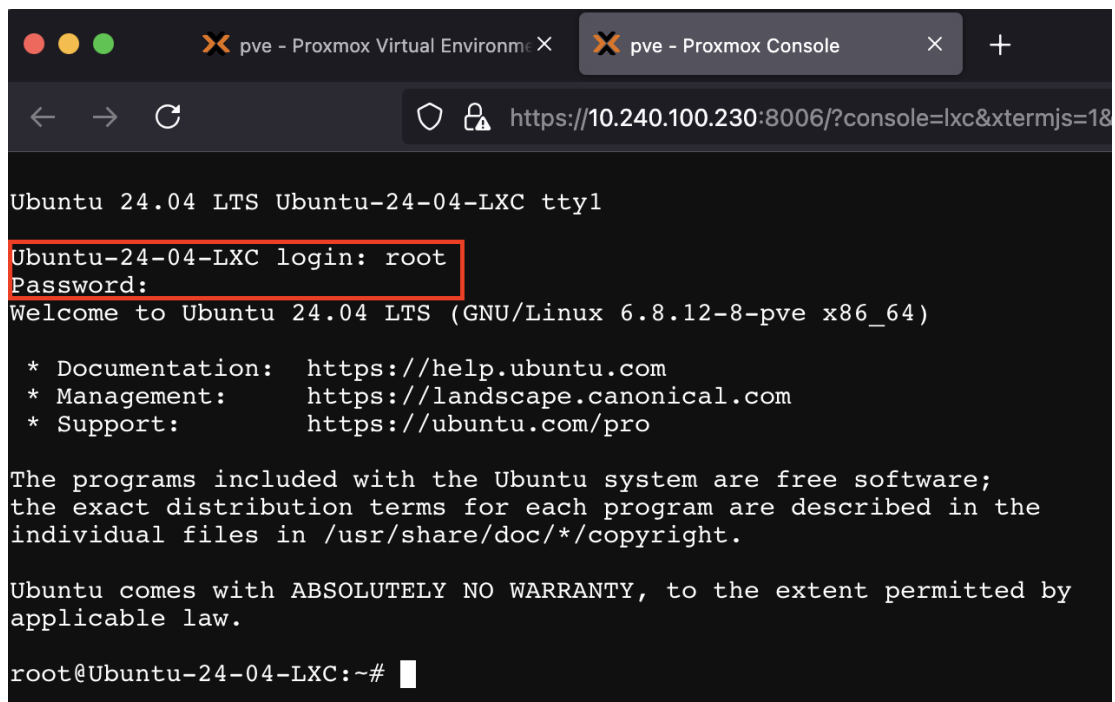
Εικόνα 49 Μενού διαχείρισης λειτουργιών εικονικών μηχανών και LXC κοντέινερ του διαχειριστικού περιβάλλοντος του Proxmox.

Για να διαχειριστούμε το λειτουργικό σύστημα της εικονικής μηχανής που επιθυμούμε αρκεί να κάνουμε δεξί κλικ πάνω στο όνομα του, από την δεξιά στήλη του διαχειριστικού περιβάλλοντος του Proxmox όπως φαίνεται στην Εικόνα 49 και να διαλέξουμε την επιλογή «Console». Στην συνέχεια θα μας εμφανιστεί ένα νέο παράθυρο στο οποίο φαίνεται η εικόνα που θα είχαμε αν είχαμε μια πραγματική οθόνη συνδεδεμένη.



Εικόνα 50 Λειτουργία κονσόλας του διαχειριστικού περιβάλλοντος του Proxmox.

Στην Εικόνα 51, απεικονίζεται ένα παράδειγμα στο οποίο συνδεόμαστε στο κοντέινερ που υλοποιήσαμε στο υποκεφάλαιο 4.2.4 μέσω της κονσόλας ώστε να το διαχειριστούμε.



```
pve - Proxmox Virtual Environm X pve - Proxmox Console
https://10.240.100.230:8006/?console=lxc&xtermjs=18
Ubuntu 24.04 LTS Ubuntu-24-04-LXC tty1
Ubuntu-24-04-LXC login: root
Password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.12-8-pve x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@Ubuntu-24-04-LXC:~#
```

Εικόνα 51 Διαχείριση ενός λειτουργικού συστήματος χρησιμοποιώντας την λειτουργία του διαχειριστικού περιβάλλοντος του Proxmox.

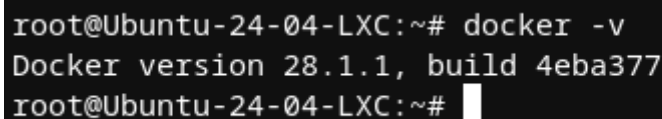
4.3 Εγκατάσταση και Ρύθμιση του Docker

Θα περιγράψουμε την διαδικασία εγκατάστασης του Docker για το λειτουργικό σύστημα «Ubuntu 24.04 LTS Server» το οποίο και εγκαταστήσαμε σε ένα LXC κοντέινερ στο υποκεφάλαιο 4.2.4. Η καλύτερη πρακτική είναι να ακολουθήσουμε τις αναλυτικές οδηγίες που αναγράφονται στην επίσημη ιστοσελίδα στον σύνδεσμο «<https://docs.docker.com/engine/install/ubuntu/>». Αυτό είναι πολύ σημαντικό διότι με τον καιρό υπάρχει πιθανότητα κάποια από τα βήματα να αλλάξουν ή να μην υποστηρίζονται πλέον. Κατά την σύνταξη της παρούσας εργασίας χρειάστηκε να εκτελέσουμε της παρακάτω εντολές στη σειρά από την κονσόλα μέσω του διαχειριστικού περιβάλλοντος του Proxmox για να εγκαταστήσουμε το Docker:

- `sudo apt update && sudo apt upgrade`
- `sudo apt install curl`
- `curl -fsSL https://get.docker.com/ | sh`

Αφού ολοκληρώσουμε την διαδικασία της εγκατάστασης για να βεβαιωθούμε ότι έχει εγκατασταθεί επιτυχώς το Docker αρκεί να τρέξουμε την παρακάτω εντολή η οποία θα πρέπει να μας επιστρέψει την εγκατεστημένη έκδοση:

- `docker -v`



```
root@Ubuntu-24-04-LXC:~# docker -v
Docker version 28.1.1, build 4eba377
root@Ubuntu-24-04-LXC:~#
```

Εικόνα 52 Επιβεβαίωση της ορθής εγκατάστασης του Docker.

Στην περίπτωση μας επέστρεψε “*Docker version 28.1.1, build 4eba377*” όπως φαίνεται στην Εικόνα 52 και συνεπώς, η εγκατάσταση έχει ολοκληρωθεί επιτυχώς.

4.4 Ανάπτυξη Kubernetes Cluster

Παρακάτω περιγράφεται η διαδικασία εγκατάστασης του K3s στην εικονική μηχανή με λειτουργικό σύστημα Ubuntu 24.04.02 που δημιουργήσαμε στο υποκεφάλαιο 4.2.3 καθώς και ο έλεγχος της λειτουργίας του cluster. Όπως και στην εγκατάσταση του Docker παραπάνω, η καλύτερη πρακτική είναι να είναι να ακολουθήσουμε τις αναλυτικές οδηγίες που αναγράφονται στην επίσημη ιστοσελίδα στον σύνδεσμο «<https://docs.k3s.io/quick-start/>». Η εγκατάσταση γίνεται με μια μόνο εντολή όπως φαίνεται παρακάτω:

- `curl -sfL https://get.k3s.io | sh -`

Όταν ολοκληρώσαμε την εγκατάσταση, χρησιμοποιήσαμε την παρακάτω εντολή ώστε να ελέγξουμε σε τι κατάσταση βρίσκεται ο cluster μας:

- `sudo kubectl get nodes`

```
dimitris@ubuntu-24-04-vm:~$ sudo kubectl get nodes
NAME                STATUS    ROLES                  AGE      VERSION
ubuntu-24-04-vm    Ready     control-plane,master   2m29s    v1.32.4+k3s1
dimitris@ubuntu-24-04-vm:~$
```

Εικόνα 53 Επιβεβαίωση της ορθής εγκατάστασης του K3s cluster.

Στην περίπτωση μας όπως φαίνεται στην Εικόνα 53, μας επέστρεψε ότι το «STATUS» είναι «Ready», δηλαδή πως η κατάσταση του cluster είναι έτοιμη για χρήση κάτι τι οποίο επιβεβαιώνει την επιτυχή εγκατάσταση.

4.5 Υλοποίηση Συστήματος Παρακολούθησης (Monitoring)

Ως λογισμικό παρακολούθησης θα χρησιμοποιήσουμε το «Gatus», το οποίο περιγράψαμε στο υποκεφάλαιο 3.3.4. Θα δείξουμε την διαδικασία εγκατάστασης και ρύθμισης του μέσα σε ένα κοντέινερ χρησιμοποιώντας το Docker, φιλοξενώντας το μέσα στο LXC κοντέινερ στο οποίο έχουμε εγκαταστήσει το Docker στο υποκεφάλαιο 4.3. Από την κονσόλα του LXC κοντέινερ, μέσω του διαχειριστικού περιβάλλοντος του Proxmox, ακολουθήσαμε την εξής διαδικασία:

Αρχικά δημιουργήσαμε έναν φάκελο με όνομα «gatus» μέσα στον οποίο θα αποθηκεύσουμε όλα τα απαραίτητα αρχεία της εφαρμογής, και στην συνέχεια μεταβήκαμε σε αυτόν χρησιμοποιώντας την εντολή:

- `mkdir gatus && cd gatus`

Έπειτα, χρησιμοποιώντας τις παρακάτω εντολές, δημιουργήσαμε δυο αρχεία, ένα με το όνομα «compose.yaml» το οποίο είναι το αρχείο οδηγιών του Docker Compose και ένα με το όνομα «config.yaml» το οποίο είναι καθορίζει τις ρυθμίσεις και τα endpoints που θα παρακολουθεί το Gatus:

Αρχείο compose.yaml:

- `cat << EOF > compose.yaml`

```

services:
  gatus:
    image: twinproduction/gatus:latest
    ports:
      - 8080:8080
    volumes:
      - ./config.yaml:/config/config.yaml
EOF

```

Επεξήγηση παραμέτρων:

- `services`: Ορίζει τα containers που θα τρέξουν υπό το ίδιο Docker Compose project.
- `gatus.image`: Χρησιμοποιεί το επίσημο πρότυπο `twinproduction/gatus:latest` που περιλαμβάνει το Gatus⁶.
- `gatus.ports`: Αντιστοιχίζει την πόρτα 8080 του μηχανήματος μας (στην περίπτωση μας του LXC κοντέινερ) στην πόρτα 8080 του Docker container, επιτρέποντας πρόσβαση στην εφαρμογή μέσω `http://<διεύθυνση_IP>:8080`.
- `gatus.volumes`: Προωθεί το τοπικό αρχείο «./config.yaml» μέσα στο Docker container στην διαδρομή «/config/config.yaml», όπου το Gatus διαβάζει το αρχείο ρυθμίσεων.

Αρχείο `config.yaml`:

- `cat << EOF > config.yaml`
`endpoints:`
 - `name: example`
 `url: https://example.org`
 `interval: 30s`
 `conditions:`
 - `"[STATUS] == 200"`
`EOF`

Επεξήγηση παραμέτρων:

- `endpoints`: Λίστα με όλα τα “checks” που επιθυμούμε να εκτελεί το Gatus⁷.
- `name`: Μοναδικό αναγνωριστικό για κάθε endpoint το οποίο εμφανίζεται στην σελίδα παρακολούθησης.
- `url`: Η διεύθυνση που θα ελέγχεται. Υποστηρίζονται HTTP, ICMP, TCP και DNS πρωτόκολλα.
- `interval`: Χρονικό διάστημα μεταξύ δύο διαδοχικών ελέγχων (π.χ. 30s, 5m). Η προεπιλεγμένη τιμή είναι 60s (δευτερόλεπτα) αν δεν οριστεί.
- `conditions`: Λίστα εκφράσεων που αξιολογούνται μετά από κάθε έλεγχο. Στο παράδειγμα, το check επιτυγχάνεται μόνο αν το HTTP status code είναι 200.

Στο παραπάνω παράδειγμα, το Gatus θα στέλνει κάθε 30 δευτερόλεπτα ένα αίτημα στο `https://example.org` και θα καταγράφει επιτυχία μόνο όταν λάβει απάντηση με status 200.

Μπορούμε να ελέγξουμε το περιεχόμενο των αρχείων που δημιουργήσαμε χρησιμοποιώντας την παρακάτω εντολή:

⁶ <https://hub.docker.com/r/twinproduction/gatus>

⁷ <https://github.com/TwiN/gatus>

- `cat < όνομα αρχείου >`

The screenshot shows a terminal window for a Docker container named 'Container 101 (Ubuntu-24-04-LXC) on node 'pve''. The terminal displays the following commands and their outputs:

```

root@Ubuntu-24-04-LXC:~# mkdir gatus && cd gatus
root@Ubuntu-24-04-LXC:~/gatus# cat << EOF > compose.yaml
services:
  gatus:
    image: twinproduction/gatus:latest
    ports:
      - 8080:8080
    volumes:
      - ./config.yaml:/config/config.yaml
EOF
root@Ubuntu-24-04-LXC:~/gatus# cat << EOF > config.yaml
endpoints:
  - name: example
    url: https://example.org
    interval: 30s
    conditions:
      - "[STATUS] == 200"
EOF
root@Ubuntu-24-04-LXC:~/gatus# cat compose.yaml
services:
  gatus:
    image: twinproduction/gatus:latest
    ports:
      - 8080:8080
    volumes:
      - ./config.yaml:/config/config.yaml
root@Ubuntu-24-04-LXC:~/gatus# cat config.yaml
endpoints:
  - name: example
    url: https://example.org
    interval: 30s
    conditions:
      - "[STATUS] == 200"
root@Ubuntu-24-04-LXC:~/gatus#

```

Εικόνα 54 Εκτέλεση εντολών προετοιμασίας εγκατάστασης του Gatus Docker κοντέινερ.

Αφού έχουμε δημιουργήσει τα παραπάνω αρχεία, για να εκκινήσουμε την εφαρμογή εκτελούμε την εντολή:

- `docker compose up -d`

Μετά την εκκίνηση μπορούμε προαιρετικά να:

- Να δούμε την κατάσταση της εφαρμογής εκτελώντας την εντολή:
 - `docker compose ps`
- Να παρακολουθήσουμε τα αρχεία καταγραφής της εφαρμογής εκτελώντας την εντολή:
 - `docker compose logs`
- Να σταματήσουμε την εφαρμογή χρησιμοποιούμε την εντολή:
 - `docker compose down`

```
root@Ubuntu-24-04-LXC::~~# gatus# docker compose up -d
[+] Running 4/4
   ✓ gatus Pulled
     ✓ 5f89c74c5712 Pull complete
     ✓ 592226da6b57 Pull complete
     ✓ 57b09ee07970 Pull complete
[+] Running 2/2
   ✓ Network gatus_default Created
   ✓ Container gatus-gatus-1 Started
root@Ubuntu-24-04-LXC::~gatus# docker compose ps
```

	NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS	PORTS
	gatus-gatus-1	twinproduction/gatus:latest	" /gatus"	gatus	5 seconds ago	Up 5 seconds	0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp

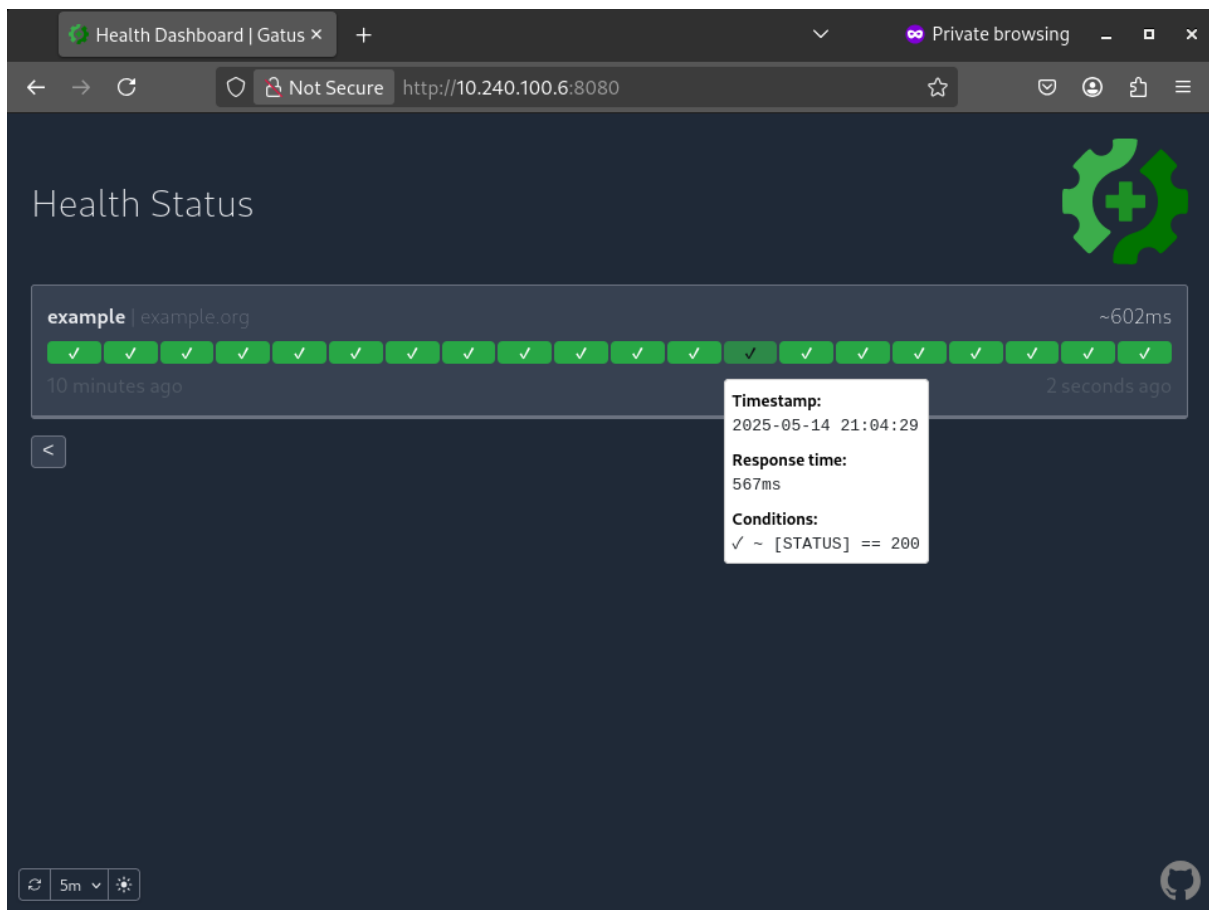
```
root@Ubuntu-24-04-LXC::~~# gatus# docker compose logs
gatus-1 | 2025/05/14 17:52:27 [main.configureLogging] Log Level is set to INFO
gatus-1 | 2025/05/14 17:52:27 [config.LoadConfiguration] Reading configuration from configFile=config/config.yaml
gatus-1 | 2025/05/14 17:52:27 [config.validateAlertingConfig] Alerting is not configured
gatus-1 | 2025/05/14 17:52:27 [config.validateEndpointsConfig] Validated 1 endpoints
gatus-1 | 2025/05/14 17:52:27 [config.validateEndpointsConfig] Validated 0 external endpoints
gatus-1 | 2025/05/14 17:52:27 [store.Initialize] Creating storage provider of type=memory
gatus-1 | 2025/05/14 17:52:27 [controller.Handle] Listening on 0.0.0.0:8080
gatus-1 | 
gatus-1 | 
gatus-1 | Fiber v2.52.6
gatus-1 | http://[::]:8080
gatus-1 | 
gatus-1 | Handlers ..... 33 Processes ..... 1
gatus-1 | Prefork ..... Disabled PID ..... 1
gatus-1 | 
gatus-1 | 2025/05/14 17:52:29 [watchdog.execute] Monitored group=: endpoint=example; key=_example; success=true; errors=0; duration=626ms
root@Ubuntu-24-04-LXC::~~# gatus# ip a s eth0 | grep -E -o 'inet [0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | cut -d'.' -f2
10.240.100.6
root@Ubuntu-24-04-LXC::~~# gatus# █
```

Εικόνα 55 Εκτέλεση εντολών για την εκκίνηση του Gatus κοντέινερ.

Όπως φαίνεται στην Εικόνα 55 η εκκίνηση του Gatus έγινε επιτυχώς και επισκέπτοντας τον σύνδεσμο http://<διεύθυνση_IP>:8080, όπου <διεύθυνση_IP> είναι η τοπική διεύθυνση IP που έχει λάβει το μηχάνημα μας, μας εμφανίζεται η σελίδα του Gatus όπως φαίνεται στην Εικόνα 56.

Ένας τρόπος να βρούμε την διεύθυνση IP σε περίπτωση που δεν την γνωρίζουμε, είναι να χρησιμοποιήσουμε την εξής εντολή η οποία θα μας επιστρέψει την διεύθυνση IP για την κάρτα δικτύου με όνομα «eth0»:

- `ip a s eth0 | grep -E -o 'inet [0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | cut -d'.' -f2`



Εικόνα 56 Η σελίδα του Gatus που υλοποιήσαμε.

5

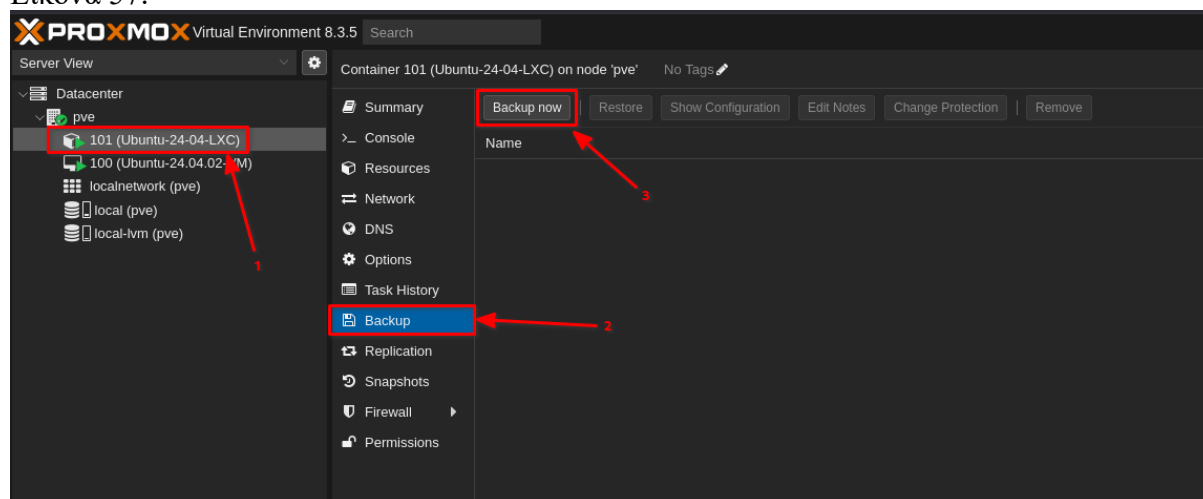
Αντίγραφα Ασφαλείας και Αξιολόγηση Απόδοσης

5.1 Αντίγραφα Ασφαλείας

Η ύπαρξη αντιγράφων ασφαλείας αποτελεί θεμελιώδη προϋπόθεση για τη διασφάλιση της ακεραιότητας και λειτουργίας κάθε υποδομής, συμπεριλαμβανομένου και ενός οικιακού εργαστηρίου homelab. Παρόλο που ένα homelab εργαστήριο δεν είναι απαραίτητο να φιλοξενεί κρίσιμες υπηρεσίες όπως για παράδειγμα θα έκανε μια υποδομή μιας εταιρίας, ο κίνδυνος ανθρώπινου λάθους, κακόβουλων επιθέσεων ή απρόβλεπτων σφαλμάτων υλικού και λογισμικού συνεχίζει να υφίσταται. Για τον λόγο αυτό, συνιστάται η τακτική λήψη αντιγράφων ασφαλείας κάθε μορφής για την αποφυγή απώλειας δεδομένων. Παρακάτω περιγράφεται ο τρόπος λήψης και επαναφοράς αντιγράφων ασφαλείας εικονικών μηχανών και LXC κοντέινερ, μέσα από το διαχειριστικό περιβάλλον του Proxmox.

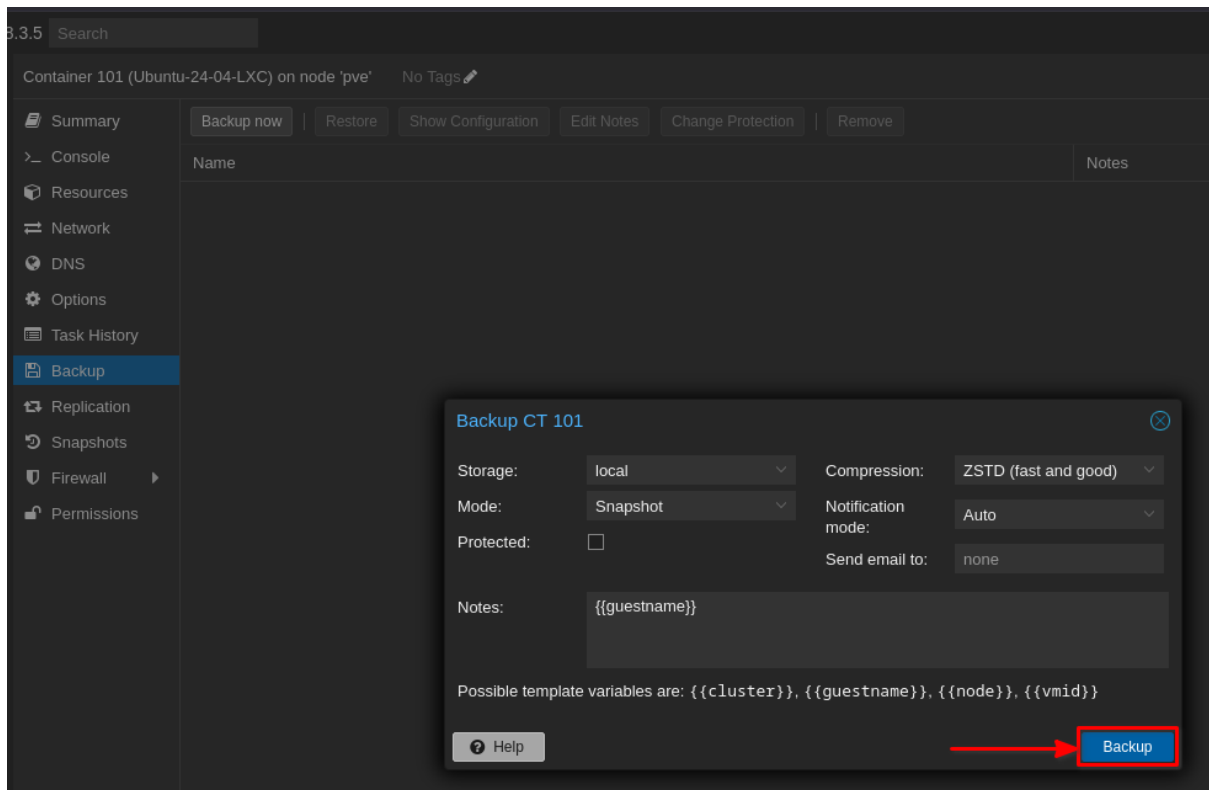
5.1.1 Δημιουργία αντίγραφου ασφάλειας στο Proxmox

Τα βήματα που ακολουθούμε για την δημιουργία ενός αντίγραφου ασφαλείας είναι τα ίδια για εικονικές μηχανές και LXC κοντέινερ. Στο παρακάτω παράδειγμα θα δημιουργήσουμε ένα αντίγραφο ασφαλείας για το LXC κοντέινερ που υλοποιήσαμε στο υποκεφάλαιο 4.2.4. Ξεκινάμε την διαδικασία από την αριστερή στήλη της κεντρικής οθόνης του διαχειριστικού περιβάλλοντος του Proxmox επιλέγοντας το LXC κοντέινερ, έπειτα από την μεσαία στήλη διαλέγουμε την επιλογή «Backup» και πατάμε το κουμπί «Backup now» όπως φαίνεται στην Εικόνα 57.



Εικόνα 57 Διαδικασία δημιουργίας αντίγραφου ασφαλείας μέσα από το διαχειριστικό περιβάλλον του Proxmox.

Από το παράθυρο που θα εμφανιστεί πατάμε το κουμπί «Backup» ώστε να ξεκινήσει η διαδικασία με την προεπιλεγμένη μέθοδο (Mode) «Snapshot» όπως φαίνεται στην Εικόνα 58.



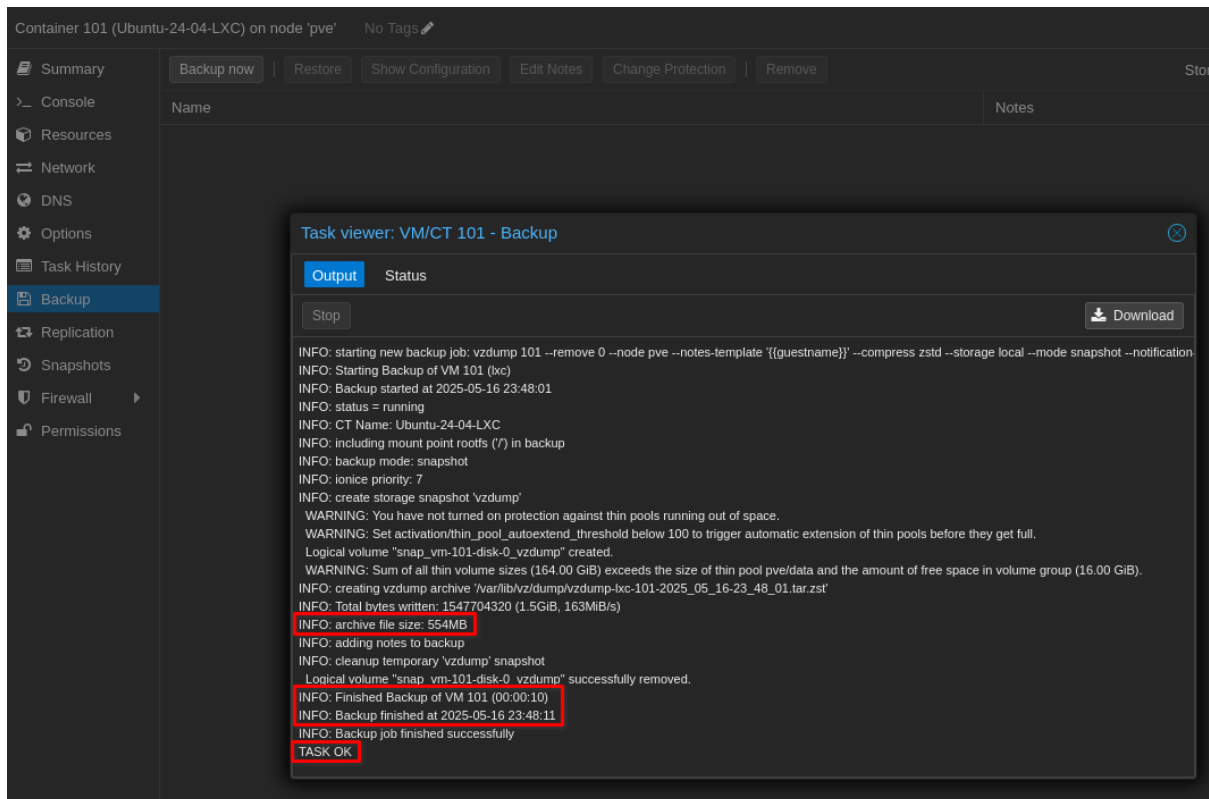
Εικόνα 58 Δημιουργία αντίγραφου ασφαλείας με την μέθοδο «Snapshot» μέσα από το διαχειριστικό περιβάλλον του Proxmox.

Σημειώνεται πως υπάρχουν τρεις μέθοδοι λήψης ενός αντίγραφου ασφαλείας στο Proxmox⁸, οι οποίες επιλέγονται ανάλογα με τις απαιτήσεις μας:

- **Snapshot:** Λειτουργία στιγμιότυπου η οποία παρέχει το χαμηλότερο χρόνο διακοπής λειτουργίας αλλά με ένα μικρό ρίσκο ασυνέπειας στα δεδομένα, λόγω του ότι εκτελεί ένα ζωντανό αντίγραφο ασφαλείας.
- **Suspend:** Λειτουργία αναστολής η οποία βάζει την μηχανή σε αναστολή πριν καλέσει την λειτουργία στιγμιότυπου. Δεν βελτιώνει απαραίτητα την συνέπεια των δεδομένων, συνεπώς ανάμεσα στα δύο συνιστάται η χρήση της λειτουργίας στιγμιότυπου.
- **Stop:** Λειτουργία διακοπής η οποία παρέχει την υψηλότερη συνέπεια στα δεδομένα αλλά με κόστος τον τερματισμό της εικονικής μηχανής ή LXC κοντέινερ κατά την διάρκεια της λήψης του αντίγραφου και την εκκίνηση του αφού ολοκληρωθεί.

Μετά την ολοκλήρωση, θα εμφανιστεί στο παράθυρο η κατάσταση, το μέγεθος και ο χρόνος λήψης του αντιγράφου όπως φαίνεται στην Εικόνα 59.

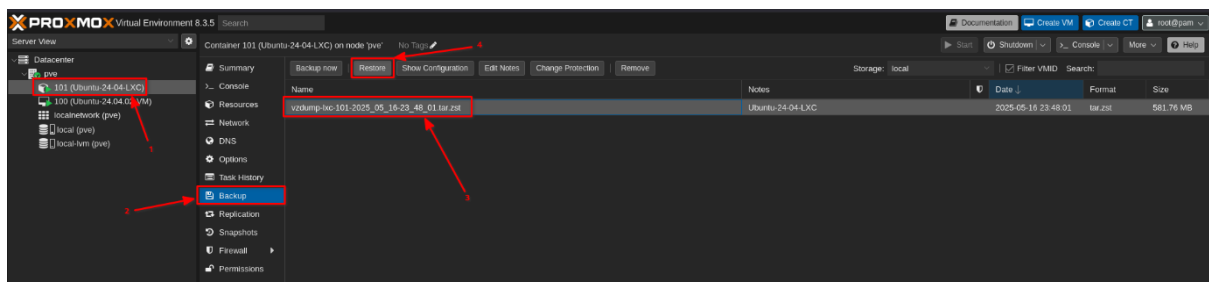
⁸ https://pve.proxmox.com/wiki/Backup_and_Restore



Εικόνα 59 Ολοκλήρωση διαδικασίας λήψης αντίγραφου ασφαλείας μέσα από το διαχειριστικό περιβάλλον του Proxmox.

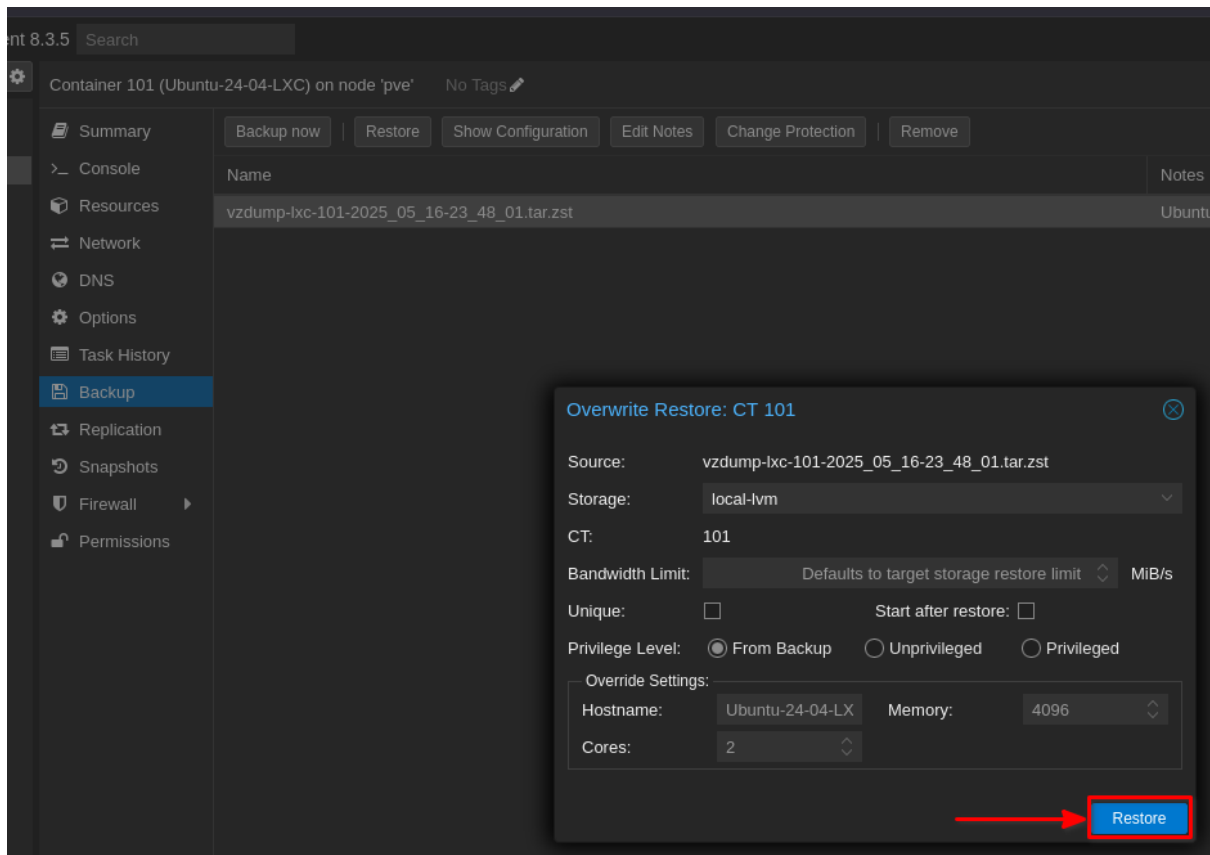
5.1.2 Επαναφορά αντίγραφου ασφαλείας στο Proxmox

Η επαναφορά ενός αντίγραφου ασφαλείας, είναι παρόμοια με την διαδικασία που αναφέραμε παραπάνω για την λήψη τους και μπορεί επίσης να γίνει με ευκολία μέσα από το διαχειριστικό περιβάλλον του Proxmox. Ξεκινάμε τη διαδικασία από την αριστερή στήλη της κεντρικής οθόνης του διαχειριστικού περιβάλλοντος του Proxmox, επιλέγοντας την εικονική μηχανή ή LXC κοντέινερ που θέλουμε να επαναφέρουμε, έπειτα από την μεσαία στήλη διαλέγουμε την επιλογή «Backup» η οποία μας εμφανίζει τα αντίγραφα ασφαλείας που υπάρχουν για την εκάστοτε μηχανή. Επιλέγουμε αυτό που θέλουμε να επαναφέρουμε και πατάμε το κουμπί «Restore» όπως φαίνεται στην Εικόνα 60.



Εικόνα 60 Διαδικασία επαναφοράς αντίγραφου ασφαλείας μέσα από το διαχειριστικό περιβάλλον του Proxmox.

Από το παράθυρο που θα εμφανιστεί ελέγχουμε τις επιλογές μας, πατάμε το κουμπί «Restore» όπως φαίνεται στην Εικόνα 61 και περιμένουμε μέχρι να ολοκληρωθεί η διαδικασία.



Εικόνα 61 Επαναφορά αντίγραφου ασφαλείας μέσα από το διαχειριστικό περιβάλλον του Proxmox.

5.2 Παρακολούθηση και Ανάλυση Χρήσης Πόρων

Παρακάτω παρουσιάζεται η σύγκριση της κατανάλωσης πόρων μεταξύ ενός LXC κοντέινερ και μιας εικονικής μηχανής βάσει των μετρήσεων για μια ώρα μέσα από το διαχειριστικό περιβάλλον του Proxmox.

5.2.1 Σύγκριση χρήσης πόρων LXC κοντέινερ και εικονικής μηχανής

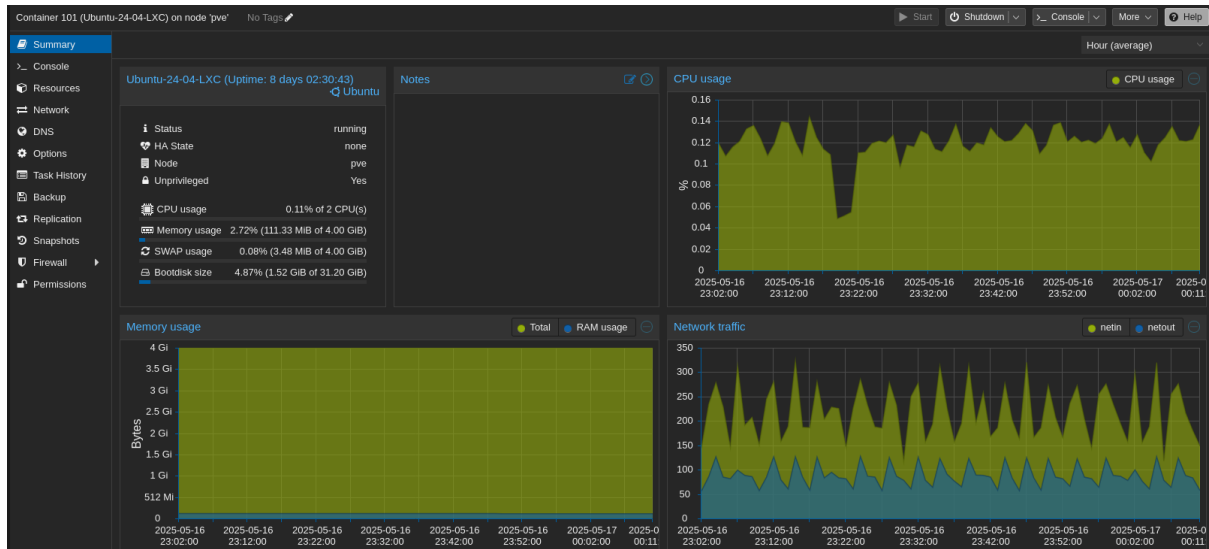
Οι μελέτες δείχνουν ότι οι εικονικές μηχανές απαιτούν σημαντικά περισσότερους πόρους συστήματος για την φιλοξενία τους σε σχέση με τα LXC κοντέινερ, καθώς φέρουν ολόκληρο λειτουργικό σύστημα (Lindström, 2022).

Για το homelab εργαστήριο που υλοποιήσαμε στην παρούσα εργασία τα ευρήματα είναι τα εξής:

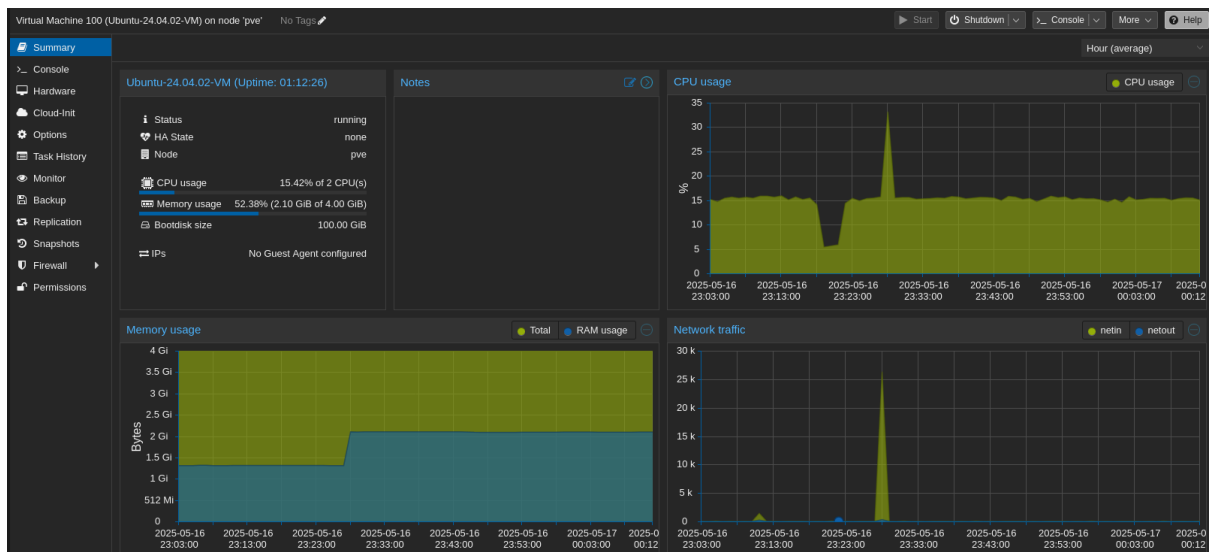
Πόροι	LXC κοντέινερ (Μέση τιμή)	Εικονική μηχανή (Μέση τιμή)
Επεξεργαστής CPU	~ 0.12%	~ 15%
Μνήμη RAM	~ 120MiB	~ 2GiB

Τα αποτελέσματα των μετρήσεων αναδεικνύουν τη σαφή υπεροχή του LXC κοντέινερ έναντι της εικονικής μηχανής όσον αφορά την αποδοτικότητα πόρων. Το LXC κοντέινερ επιφέρει μέση κατανάλωση πόρων επεξεργαστή μόλις 0,12%. Αντίθετα, η εικονική μηχανή εμφανίζει

μέση κατανάλωση πόρων επεξεργαστή περί του 15%, αποτέλεσμα που οφείλεται στην ανάγκη διαχείρισης ενός πλήρους λειτουργικού συστήματος. Σε ό,τι αφορά τη μνήμη RAM, το LXC κοντέινερ απαιτεί περίπου 120 MiB, ενώ η εικονική μηχανή, καταναλώνει γύρω στα 2 GiB RAM πάλι λόγω της φόρτωσης ολόκληρου λειτουργικού συστήματος, αυξάνοντας κατά πολύ την κατανάλωση πόρων. Στην Εικόνα 62 και Εικόνα 63 φαίνονται οι μετρήσεις μιας ώρας μέσα από το διαχειριστικό περιβάλλον του Proxmox.



Εικόνα 62 Κατανάλωση πόρων συστήματος του LXC κοντέινερ που υλοποιήθηκε στο υποκεφάλαιο 4.2.4.

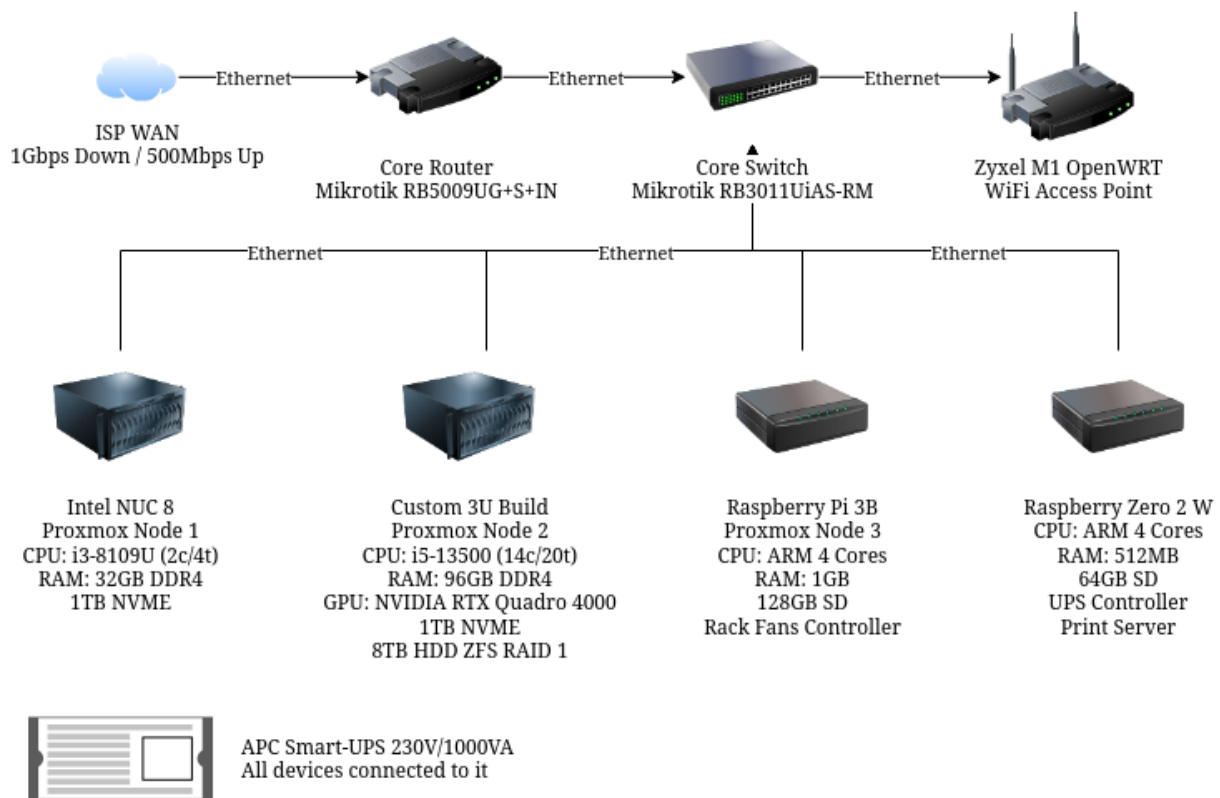


Εικόνα 63 Κατανάλωση πόρων συστήματος της εικονικής μηχανής που υλοποιήθηκε στο υποκεφάλαιο 4.2.3.

Παρουσίαση Προσωπικού Homelab

Σε αυτό το κεφάλαιο θα παρουσιάσω το προσωπικό μου homelab, το οποίο αποτελεί ένα εργαστήριο πληροφορικής που βρίσκεται στο σπίτι μου και λειτουργεί ως πεδίο πειραματισμού, εκμάθησης και αυτοματοποίησης. Μέσα από το homelab μου, έχω την δυνατότητα να υλοποιώ υπηρεσίες που διευκολύνουν την καθημερινότητα μου, να δοκιμάζω νέες τεχνολογίες και να βελτιώνω τις γνώσεις μου στον χώρο της πληροφορικής.

6.1 Διάγραμμα Υποδομής



Εικόνα 64 Σχεδιάγραμμα του προσωπικού μου homelab.

6.2 Γενική Περιγραφή

Το homelab μου είναι σχεδιασμένο ώστε να λειτουργεί αδιάκοπα 24/7, με όσο το δυνατόν χαμηλότερη κατανάλωση ενέργειας καθώς και τον λιγότερο θόρυβο. Ο εξοπλισμός που έχω επιλέξει, συνδυάζει αρκετά καλή σχέση τιμής με την απόδοση, αξιοπιστία και κατανάλωση ενέργειας. Το μεγαλύτερο μέρος του υλικού βρίσκεται συγκεντρωμένο σε ένα Rack μεγέθους 12U και το υπόλοιπο σε διάφορα σημεία του σπιτιού στα οποία ήταν εφικτό λόγω μειωμένου χώρου. Επίσης, όλα τα συστήματα τροφοδοτούνται από ένα κεντρικό UPS, ώστε το ενδεχόμενο των διακοπών ρεύματος να είναι αρκετά χαμηλότερο, το οποίο βοηθά τις

υπηρεσίες να έχουν μεγαλύτερη διαθεσιμότητα και στην γενική υγεία του υλικού μειώνοντας έτσι τον κίνδυνο απώλειας δεδομένων σε περίπτωση απροόπτου τερματισμού.

6.2.1 Υπολογιστική Υποδομή

Κόμβος 1: Intel NUC 8

Έχω επιλέξει τον συγκεκριμένο κόμβο λόγω του ότι είναι ενεργειακά αποδοτικός, ιδανικός για βασικές υπηρεσίες και εργασίες που δεν απαιτούν υψηλή υπολογιστική ισχύ. Το λειτουργικό σύστημα του είναι Proxmox και είναι ο πρώτος κόμβος του Proxmox Cluster.

Τεχνικά χαρακτηριστικά:

- 4 πυρήνες CPU
- 32GB μνήμη RAM
- 1TB σκληρό δίσκο τύπου NVME.

Είναι ιδανικό για υπηρεσίες όπως το Gatus το οποίο παρουσιάζεται στο υποκεφάλαιο 3.3.4, αλλά και λογισμικά όπως το Home Assistant.

Κόμβος 2: Custom 3U Server

Ο συγκεκριμένος κόμβος έχει σχεδιαστεί για βαριές εργασίες όπως μεγαλύτερες εικονικές μηχανές με GPU passthrough για δοκιμές μοντέλων ΑΙ και Cloud Gaming. Ακόμη έχει αρκετό αποθηκευτικό χώρο για υπηρεσίες συγχρονισμού αρχείων, αντίγραφα ασφάλειας και πολυμέσα. Ο συγκεκριμένος κόμβος επίσης έχει λειτουργικό σύστημα Proxmox και είναι ο 2ος κόμβος του Proxmox Cluster. Το 3U δηλώνει τον χώρο που καταναλώνει στο Rack σε μονάδα μέτρησης Rack Units (U).

Τεχνικά χαρακτηριστικά:

- 20 πυρήνες CPU
- 96GB μνήμη RAM
- Κάρτα γραφικών με 8GB μνήμη RAM
- 1 TB σκληρό δίσκο τύπου NVME
- 8TB σκληρό δίσκο τύπου HDD RAID 1

Η τεχνολογία RAID 1 αντικατοπτρίζει αυτόματα τα δεδομένα μεταξύ 2 μονάδων δίσκου ίδιας χωρητικότητας και μας προσφέρει ουσιαστικά 2 αντίγραφα των δεδομένων μας. Αν δηλαδή χαλάσει ο ένας δίσκος οι υπηρεσίες θα συνεχίσουν να λειτουργούν απροβλημάτιστα διότι τα δεδομένα θα υπάρχουν στον άλλο.

Κόμβος 3: Raspberry Pi 3B

Η βασική λειτουργία του συγκεκριμένου κόμβου είναι η διαχείριση των ανεμιστήρων μέσα στο Rack με βάση την θερμοκρασία που έχει την δεδομένη στιγμή. Είναι μια ειδική

κατασκευή προσαρμοσμένη στις ανάγκες μου χρησιμοποιώντας την γλώσσα προγραμματισμού Python, ανεμιστήρες ηλεκτρονικού υπολογιστή και έναν αισθητήρα θερμοκρασίας και υγρασίας. Ακόμη συλλέγει στατιστικά για την λειτουργία του όπως θερμοκρασία, ταχύτητα περιστροφής των ανεμιστήρων και στέλνει ειδοποιήσεις σε περίπτωση που σταματήσουν οι ανεμιστήρες ή ανεβεί η θερμοκρασία σε πολύ ψηλά επίπεδα. Η δεύτερη λειτουργία του είναι να συμμετέχει ως 3ος κόμβος στον Proxmox Cluster χωρίς όμως να φιλοξενεί κάποια υπηρεσία, ώστε να καλύπτει τις ελάχιστες απαιτήσεις που ορίζει το Proxmox για την δημιουργία ενός παραγωγικού Cluster.

Τεχνικά χαρακτηριστικά:

- 4 πυρήνες CPU
- 1GB μνήμη RAM
- 128GB κάρτα SD

Κόμβος 4: Raspberry Pi Zero 2 W

Ο κόμβος αυτός έχει σχεδιαστεί για να επιτρέπει σε ένα παλαιότερο UPS το οποίο δεν διαθέτει δικτυακές ή έξυπνες λειτουργίες, να μπορεί να παρακολουθείται και να διαχειρίζεται μέσω δικτύου. Αυτό μας επιτρέπει την συλλογή στατιστικών για την κατάσταση της ηλεκτροδότησης, την κατανάλωση των συσκευών σε πραγματικό χρόνο, καθώς και τον υπολειπόμενο χρόνο λειτουργίας σε κατάσταση μπαταρίας. Επίσης στέλνει ειδοποιήσεις σε περίπτωση διακοπής ρεύματος ή υπέρτασης και τερματίζει με ασφάλεια όλους τους κόμβους όταν σε κατάσταση μπαταρίας ο υπολειπόμενος χρόνος είναι σε κρίσιμο στάδιο.

Τεχνικά χαρακτηριστικά:

- 4 πυρήνες CPU
- 512MB μνήμη RAM
- 32GB κάρτα SD

6.2.2 Δικτυακή Υποδομή

Δρομολογητής δικτύου Mikrotik RB5009UG+S+IN (Router):

Έχω επιλέξει τον συγκεκριμένο δρομολογητή λόγω της αξιοπιστίας, απόδοσης και ασφάλειας που παρέχει μέσω του Router OS λειτουργικού συστήματος που χρησιμοποιεί. Πάνω σε αυτό έχω δημιουργήσει διάφορα υποδίκτυα (VLANs) ώστε να διαχωρίζω τις συσκευές στην υποδομή μου για μεγαλύτερη ασφάλεια. Για παράδειγμα, το υποδίκτυο που βρίσκονται οι κόμβοι που αναφέρονται παραπάνω δεν έχει πρόσβαση στο διαδίκτυο για λόγους ασφάλειας, παρόλα αυτά το οικιακό δίκτυο χρειάζεται να έχει πρόσβαση στο διαδίκτυο οπότε βρίσκεται και σε διαφορετικό υποδίκτυο. Επίσης έχω υλοποιήσει έναν VPN server ώστε να μπορώ να συνδέομαι απομακρυσμένα στην υποδομή μου με ασφάλεια από οπουδήποτε βρίσκομαι.

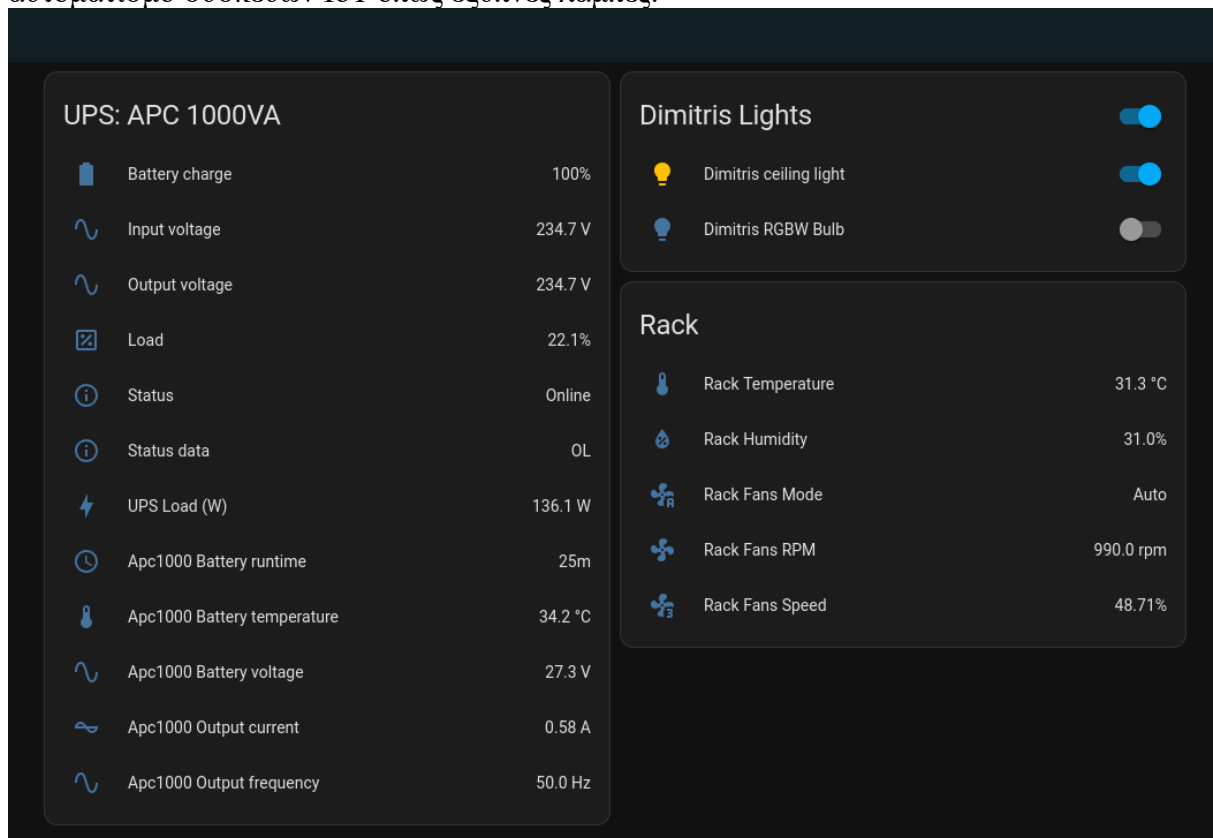
Μεταγωγέας δικτύου (Switch) και Ασύρματο σημείο πρόσβασης (Access Point):

Έχω επιλέξει την χρήση switch και access point ως ξεχωριστών συσκευών διότι προσφέρουν μια καθαρότερη αρχιτεκτονική, βελτιωμένη απόδοση, ευελιξία τοποθέτησης και προηγμένες λειτουργίες δικτύου. Με αυτόν τον τρόπο κάθε συσκευή αναλαμβάνει το ειδικό της έργο, διασφαλίζοντας την αξιοπιστία και την επεκτασιμότητα του δικτύου.

6.2.3 Υπηρεσίες που φιλοξενώ

Παρακάτω αναφέρονται κάποιες από τις κυρίες υπηρεσίες που φιλοξενώ στο προσωπικό μου homelab και μια σύντομη περιγραφή τους:

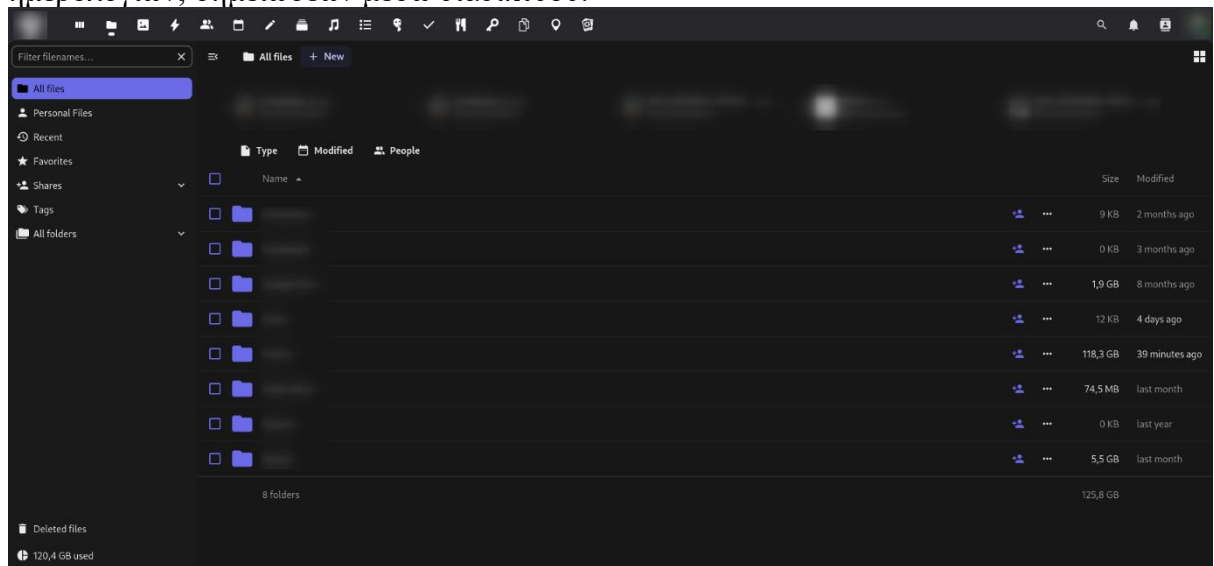
- NAS Server: Κεντρικός αποθηκευτικός χώρος για διαμοιρασμό αρχείων και πολυμέσων μέσω δικτύου.
- Home Assistant: Ένα λογισμικό ανοιχτού κώδικα για την παρακολούθηση, έλεγχο και αυτοματισμό συσκευών IoT όπως έξυπνες λάμπες.



Εικόνα 65 Στιγμιότυπο από την υπηρεσία Home Assistant που φιλοξενώ στο προσωπικό μου homelab.

Στην Εικόνα 65 στον αριστερό πίνακα φαίνεται η ενσωμάτωση που έχει γίνει με τον Κόμβο 4 από τον οποίο το Home Assistant λαμβάνει στατιστικά σε πραγματικό χρόνο για την λειτουργία του UPS. Στον δεξιό επάνω πίνακα φαίνονται οι διακόπτες για τον απομακρυσμένο έλεγχο του φωτισμού και στον κάτω πίνακα φαίνεται η ενσωμάτωση που έχει γίνει με τον Κόμβο 3, από τον οποίο το Home Assistant λαμβάνει στατιστικά θερμοκρασίας και υγρασίας του rack που βρίσκεται ο εξοπλισμός καθώς και την ταυτότητα που λειτουργούν οι ανεμιστήρες για την ροή αέρα μέσα στο rack. Το πρόγραμμα που ελέγχει τους ανεμιστήρες στο rack έχει υλοποιηθεί σε Python 3 και ο κώδικας βρίσκεται στο 9.1 Παράρτημα 1.

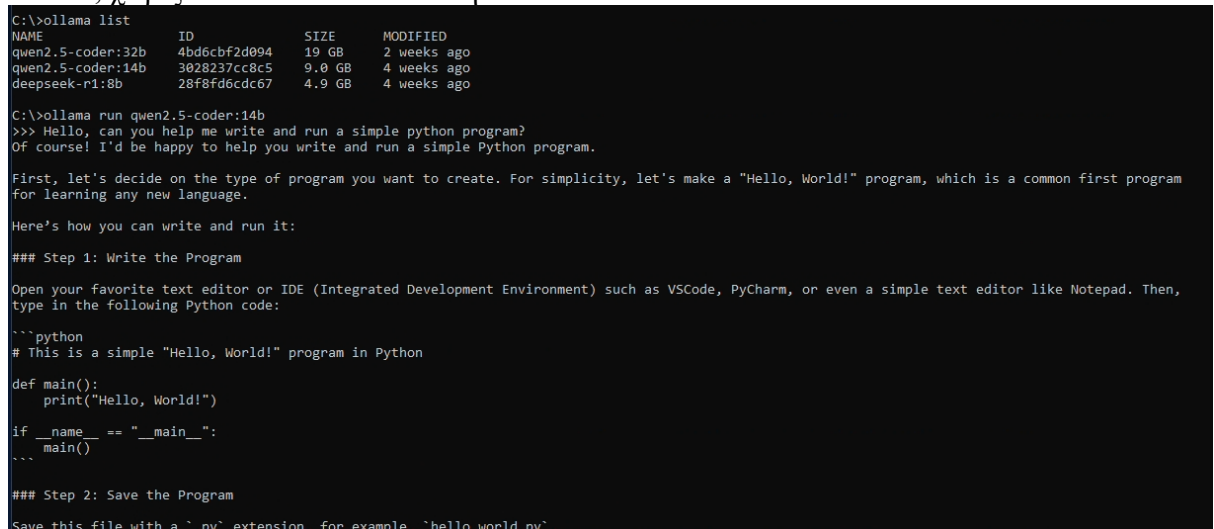
- NextCloud: Ένα λογισμικό ανοιχτού κώδικα, το οποίο προσφέρει ένα προσωπικό cloud όμοιο με τα Google Drive και Microsoft OneDrive, για συγχρονισμό αρχείων, ημερολογίων, σημειώσεων μέσω διαδικτύου.



Εικόνα 66 Στιγμιότυπο από την υπηρεσία Nextcloud που φιλοξενώ στο προσωπικό μου homelab.

Την συγκεκριμένη υπηρεσία την χρησιμοποιώ κυρίως για να δημιουργώ τακτικά αντίγραφα ασφαλείας όλων των αρχείων από το κινητό μου, καθώς και τον διαμοιρασμό αρχείων με ευκολία.

- Ollama: Ένα λογισμικό ανοιχτού κώδικα, που επιτρέπει την εκτέλεση μοντέλων (LLMs) τοπικά, χωρίς να απαιτείται σύνδεση στο διαδίκτυο.



Εικόνα 67 Στιγμιότυπο από την υπηρεσία Ollama που φιλοξενώ στο προσωπικό μου homelab.

Στην Εικόνα 67 φαίνεται ένα παράδειγμα χρήσης του ollama όπου εκτελεί τοπικά το μοντέλο Qwen2.5-Coder με 14 δισεκατομμύρια παραμέτρους, το οποίο είναι ένα εξειδικευμένο μεγάλο γλωσσικό μοντέλο που σχεδιάστηκε για εργασίες σχεδίασης, ανάλυσης και διόρθωσης κώδικα (Team, 2024).

6.2.4 Φωτογραφίες Υποδομής



Εικόνα 68 Ο εξοπλισμός του homelab μου μέσα στο Rack.

Στην Εικόνα 68 φαίνεται η βασική υποδομή του homelab μου. Στην επάνω αριστερή γωνιά μέσα σε μια εκτυπωμένη θήκη, βρίσκεται η Raspberry Pi 3B (Κόμβος 3). Κάτω από αυτή υπάρχει ο δρομολογητής δικτύου Mikrotik RB5009UG+S+IN, ακολουθούμενος από τον μεταγωγέα δικτύου Mikrotik RB3011UiAS-RM. Στην συνέχεια από κάτω φαίνεται ένας διακομιστής ο οποίος δεν χρησιμοποιείται και τέλος κάτω από αυτόν φαίνεται ο Custom 3U Server (Κόμβος 2).



Εικόνα 69 Ο κόμβος 1 του homelab μου.

Στην Εικόνα 69 φαίνεται το Intel NUC 8 ο οποίος είναι ο Κόμβος 1 και βρίσκεται στο γραφείο μου.

6.3 Χρήσιμες Διαδικασίες

6.3.1 SSH (Secure Shell)

Το SSH (Secure Shell) είναι ένα πρωτόκολλο δικτύου που προσφέρει κρυπτογραφημένη επικοινωνία μεταξύ ενός client (τοπικού υπολογιστή) και ενός server (Kaplarevic, 2024). Η σύνδεση μέσω SSH αποτελεί βασική δεξιότητα για τη διαχείριση ενός homelab, καθώς επιτρέπει την ασφαλή, απομακρυσμένη πρόσβαση και διαχείριση servers μέσω γραμμής εντολών. Παρακάτω περιγράφονται τα βασικά βήματα και οι επιλογές για τη χρήση του SSH.

1. Επιλογή λογισμικού SSH Client

- Στα Linux και macOS: Η εντολή ssh είναι προεγκατεστημένη στο τερματικό.
- Στα Windows: Μπορεί να χρησιμοποιηθεί το PuTTY ή το ενσωματωμένο OpenSSH client μέσω PowerShell ή Command Prompt.

2. Βασική Σύνταξη Εντολής

Η βασική μορφή της εντολής είναι:

```
ssh [όνομα_χρήστη]@[διεύθυνση_IP_ή_όνομα_server]
```

Για παράδειγμα:

```
ssh dimitris@10.240.100.100
```

3. Διαδικασία Σύνδεσης

- Ανοίγετε το τερματικό ή το πρόγραμμα SSH client.
- Πληκτρολογείτε την εντολή σύνδεσης.
- Την πρώτη φορά, θα σας ζητηθεί να αποδεχτείτε το αποτύπωμα (fingerprint) του server.
- Εισάγετε τον κωδικό πρόσβασης όταν ζητηθεί (τα γράμματα δεν εμφανίζονται για λόγους ασφαλείας).

6.3.2 Proxmox GPU Passthrough

Το GPU passthrough στο Proxmox επιτρέπει την απευθείας εκχώρηση μιας κάρτας γραφικών σε μια εικονική μηχανή, ώστε να αξιοποιεί πλήρως τις δυνατότητές της. Η διαδικασία απαιτεί συγκεκριμένες ρυθμίσεις στο υλικό και το λειτουργικό σύστημα του host και μπορεί να διαφέρει ανάλογα το υλικό. Η διαδικασία που θα παρουσιάσω απευθύνεται σε επεξεργαστές Intel και κάρτες γραφικών NVIDIA.

Προαπαιτούμενα:

- Υποστήριξη τεχνολογίας VT-d (Intel) ή AMD-Vi/IOMMU (AMD) στο BIOS/UEFI.
- Ενεργοποίηση τεχνολογιών IOMMU, SR-IOV.
- Η κάρτα γραφικών που θα γίνει passthrough να μην χρησιμοποιείται από το Proxmox για δίκτες του ανάγκες.

Διαδικασία:

1. Θα πρέπει να ακολουθήσετε τις οδηγίες που αναγράφονται στο εγχειρίδιο χρήσης της μητρικής κάρτας του υπολογιστή σας για την ενεργοποίηση IOMMU στο BIOS.
2. Τροποποίηση του GRUB bootloader στο Proxmox
 - Ανοίξτε το αρχείο ρυθμίσεων του GRUB: `nano /etc/default/grub`
 - Εντοπίστε τη γραμμή: `GRUB_CMDLINE_LINUX_DEFAULT="quiet"`
 - Τροποποιήστε ως εξής: `GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on iommu=pt"`
 - Αποθηκεύστε και κλείστε το αρχείο: `Ctrl+O`, `Enter`, `Ctrl+X`
 - Ενημερώστε το GRUB: `update-grub`
3. Φόρτωση των απαραίτητων VFIO modules στο Proxmox
 - Επεξεργαστείτε το αρχείο modules: `nano /etc/modules`
 - Προσθέστε τις γραμμές:
 - i. `vfio`
 - ii. `vfio_iommu_type1`
 - iii. `vfio_pci`
 - Αποθηκεύστε και κλείστε το αρχείο: `Ctrl+O`, `Enter`, `Ctrl+X`

4. Ενεργοποίηση unsafe interrupts και ειδικών επιλογών KVM
 - Εκτελέστε: `echo "options kvm ignore_msrs=1 report_ignored_msrs=0" > /etc/modprobe.d/kvm.conf`
5. Blacklist των drivers της GPU από το host
 - Εκτελέστε:
 - i. `echo "blacklist nouveau" >> /etc/modprobe.d/blacklist.conf`
 - ii. `echo "blacklist nvidia" >> /etc/modprobe.d/blacklist.conf`
 - iii. `echo "blacklist nvidiafb" >> /etc/modprobe.d/blacklist.conf`
 - iv. `echo "blacklist nvidia_drm" >> /etc/modprobe.d/blacklist.conf`
6. Προσθήκη της GPU στο VFIO
 - Βρείτε τα vendor IDs της GPU και του audio controller (π.χ. με `lspci -nn`).
 - Εκτελέστε (Αντικαθιστώντας τα IDs με τα δικά σας): `echo "options vfio-pci ids=10de:1b81,10de:10f0" > /etc/modprobe.d/vfio.conf`
7. Ενημέρωση του initramfs και επανεκκίνηση
 - Εκτελέστε:
 - i. `update-initramfs -u`
 - ii. `reboot`
8. Ρυθμίσεις στην εικονική μηχανή στο Proxmox
 - Δημιουργήστε ή επεξεργαστείτε την εικονική μηχανή που θέλετε να εκχωρήσετε την κάρτα γραφικών.
 - Προσθέστε το PCI Device (την GPU) από το Hardware tab.
 - Επιλέξτε "All Functions", "Primary GPU" και "PCI-Express" αν είναι διαθέσιμα.
 - Για Windows VM, ορίστε Machine type σε "q35" και CPU type σε "host".

7

Συμπεράσματα

Στο πλαίσιο αυτής της εργασίας, εξερευνήθηκαν και παρουσιάστηκαν οι βασικές τεχνολογίες και εργαλεία για τη δημιουργία και τη διαχείριση ενός homelab εργαστηρίου, με έμφαση στην εγκατάσταση και τη χρήση του Proxmox, του Docker και του Kubernetes. Η διαδικασία εγκατάστασης και ρύθμισης αυτών των πλατφορμών, όπως περιγράφεται αναλυτικά, αποδεικνύει ότι με τις σωστές γνώσεις και μεθοδολογίες είναι δυνατή η ανάπτυξη ενός ισχυρού οικιακού εργαστηρίου, το οποίο μπορεί να υποστηρίξει τη δοκιμή και εκτέλεση πολλών εφαρμογών ταυτόχρονα. Το Proxmox επιλέχθηκε ως βάση για τη διαχείριση των πόρων του homelab, προσφέροντας ευελιξία και δυνατότητες για τη δημιουργία εικονικών μηχανών και LXC κοντέινερ. Μέσα από την εργασία, έγινε σαφές ότι οι εικονικές μηχανές και τα κοντέινερ έχουν διαφορετικά πλεονεκτήματα και περιορισμούς, ενώ η επιλογή εξαρτάται από τις ανάγκες του χρήστη και τη φύση των εφαρμογών που θέλει να εκτελέσει.

Επιπλέον, η χρήση του Docker και του Kubernetes επέτρεψε την εύκολη ανάπτυξη, δοκιμή και παρακολούθηση εφαρμογών στο homelab. Η εγκατάσταση του Gatus σε ένα Docker κοντέινερ αποτέλεσε ένα πρακτικό παράδειγμα για την παρακολούθηση της διαθεσιμότητας και της υγείας των υπηρεσιών, ενισχύοντας την αυτοματοποίηση και τη συντήρηση του συστήματος. Η παρουσίαση του προσωπικού homelab και των εφαρμογών που εκτελούνται σε αυτό, δίνει μια πλήρη εικόνα των δυνατοτήτων και των προκλήσεων που συναντάει κάποιος στην πράξη, καθώς και των λύσεων που μπορούν να εφαρμοστούν για την αποτελεσματική διαχείριση ενός τέτοιου περιβάλλοντος. Αυτό το περιβάλλον επιτρέπει την εξοικείωση με σύγχρονες πρακτικές ανάπτυξης λογισμικού και δικτύων, ενώ ταυτόχρονα βοηθά στην απόκτηση πολύτιμων δεξιοτήτων για το μέλλον.

Τέλος, η εργασία υπαγορεύει ότι η χρήση αυτών των τεχνολογιών αποτελεί ένα ισχυρό εργαλείο για την εκμάθηση και την πρακτική εφαρμογή σύγχρονων τεχνικών εικονικοποίησης και κοντέινερ. Οι γνώσεις που αποκτήθηκαν μπορούν να εφαρμοστούν τόσο σε προσωπικό όσο και σε επαγγελματικό επίπεδο και αποτελούν μια ουσιαστική επένδυση για κάθε φοιτητή ή επαγγελματία του χώρου της πληροφορικής. Αυτό το εργαστήριο δεν είναι μόνο ένα εργαλείο εκμάθησης, αλλά και μια βάση για πειραματισμό, ανάπτυξη και εφαρμογή νέων τεχνολογιών, ετοιμάζοντας τους χρήστες για τις προκλήσεις της σύγχρονης IT βιομηχανίας.

8

Βιβλιογραφία – Πηγές

- Campanale, P. (2025, 05 4). *What Is a Homelab, and How Do You Start One?* Ανάκτηση από How-To Geek: <https://www.howtogeek.com/what-is-a-homelab-and-how-do-you-start-one/>
- Golden, B. (2007). *Virtualization For Dummies*. Hoboken, New Jersey: Wiley Publishing, Inc.
- Kaplarevic, V. (2024, October 24). *How to Connect to a Remote Server via SSH from Windows, Linux, or Mac*. Ανάκτηση από phoenixNAP: <https://phoenixnap.com/kb/ssh-to-connect-to-remote-server-linux-or-windows>
- Lindström, M. (2022, June). *Containers & Virtual machines A performance, resource & power consumption*. Ανάκτηση από DiVA Consortium: <https://www.diva-portal.org/smash/get/diva2:1665606/FULLTEXT01.pdf>
- Lowe, S. D. (2017). *Containers For Dummies®, HPE and Docker Special Edition*. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Proxmox Prepare Installation Media*. (2023, November 23). Ανάκτηση από Proxmox VE: https://pve.proxmox.com/wiki/Prepare_Installation_Media
- Team, Q. (2024, September 19). *Qwen2.5: A Party of Foundation Models!* Ανάκτηση από Qwen: <https://qwenlm.github.io/blog/qwen2.5/>
- Tonello, J. S. (2022). *Practical Linux DevOps*. Baldwinville, NY, USA: Apress.

9

Παραρτήματα

9.1 Παράρτημα 1

Ο παρακάτω κώδικας Python 3 υλοποιεί ένα σύστημα αυτόματου ελέγχου ταχύτητας PWM ανεμιστήρα υπολογιστή μέσω Raspberry Pi. Βασίζεται σε μετρήσεις θερμοκρασίας από αισθητήρα (που αποθηκεύονται σε αρχείο) και προσφέρει δύο λειτουργίες:

1. Αυτόματη λειτουργία: Ρυθμίζει δυναμικά την ταχύτητα (MIN_SPEED, MAX_SPEED) ανάλογα με τη θερμοκρασία, χρησιμοποιώντας τα όρια (TEMP_MIN, TEMP_MAX).
2. Χειροκίνητη λειτουργία: Επιτρέπει τον χειρισμό της ταχύτητας μέσω αρχείου ρυθμίσεων (/tmp/fan_speed).

Είναι σημαντικό να ορίσουμε στην μεταβλητή «PWM_PIN» την ακίδα GPIO στην οποία έχει συνδεθεί ο ανεμιστήρας στην Raspberry Pi. Για την εναλλαγή μεταξύ των λειτουργιών αρκεί να γράψουμε στο αρχείο «/tmp/fan_mode» «auto» ή «manual» αντίστοιχα. Στην περίπτωση χειροκίνητης λειτουργίας στο αρχείο «/tmp/fan_speed» γράφουμε την ταχύτητα σε ποσοστό % (π.χ. 68) και το πρόγραμμα το αναγνωρίζει αυτόματα και θέτει την επιθυμητή ταχύτητα.

```
import os
import time
import pigpio
import logging

# GPIO pin for PWM signal
PWM_PIN = 18

# Fan speed settings (0-100%)
MIN_SPEED = 35 # Minimum fan speed (%)
MAX_SPEED = 100 # Maximum fan speed (%)

# Temperature thresholds (°C)
TEMP_MIN = 29 # Minimum temperature to turn on the fan
TEMP_MAX = 40 # Temperature for full fan speed

# Fan behavior
ALWAYS_ON = True # If True, the fan will always run at MIN_SPEED or higher

# Log file location
LOG_FILE = "/var/log/fan_control.log"

# Mode and speed control files
```

```

MODE_FILE = "/tmp/fan_mode" # Contains "auto" or "manual"
SPEED_FILE = "/tmp/fan_speed" # Contains manual fan speed (0-100)

# Configure logging
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s - %(message)s",
    datefmt="%Y-%m-%d %H:%M:%S",
    handlers=[
        logging.FileHandler(LOG_FILE),
        #logging.StreamHandler() # Optional: Print logs to the console as well
    ]
)

# Initialize pigpio
pi = pigpio.pi()
if not pi.connected:
    logging.error("Error: pigpio daemon is not running. Start it with 'sudo pigpiod'.")
    exit(1)

# Set PWM frequency (25 kHz for most PWM fans)
PWM_FREQUENCY = 25000
pi.set_mode(PWM_PIN, pigpio.OUTPUT)
pi.set_PWM_frequency(PWM_PIN, PWM_FREQUENCY)

# Smoothing parameters
SMOOTHING_WINDOW = 5 # Number of readings to average
last_speeds = [] # Store recent speeds for averaging

# Track the last fan speed
last_logged_speed = None

def get_temperature():
    """Reads the rack temperature from the system."""
    try:
        with open("/root/sensors/temp_c", "r") as f:
            content = f.read().strip()
            if not content: # Check if the file is empty
                logging.error("Error: Temperature file is empty.")
                return None

            temp = float(content) # Convert the read value to a float
            return temp
    except FileNotFoundError:
        logging.error("Error: Could not read temperature file.")
        return None
    except ValueError:
        logging.error("Error: Unable to convert temperature file content to float. Content was: {}".format(content))

```

```

return None

def calculate_fan_speed(temp):
    """Calculate fan speed as a percentage based on rack temperature."""
    if temp < TEMP_MIN:
        return MIN_SPEED if ALWAYS_ON else 0 # Respect ALWAYS_ON setting
    elif temp >= TEMP_MAX:
        return MAX_SPEED
    else:
        # Scale speed between MIN_SPEED and MAX_SPEED
        scale = (temp - TEMP_MIN) / (TEMP_MAX - TEMP_MIN)
        return MIN_SPEED + (MAX_SPEED - MIN_SPEED) * scale

def set_fan_speed(speed):
    """Set the fan speed using PWM."""
    duty_cycle = int(255 * (speed / 100)) # Convert percentage to 0-255 range
    pi.set_PWM_dutycycle(PWM_PIN, duty_cycle)

def smooth_speed(speed):
    """Smooth the fan speed using a moving average."""
    global last_speeds
    last_speeds.append(speed)
    if len(last_speeds) > SMOOTHING_WINDOW:
        last_speeds.pop(0)
    return sum(last_speeds) / len(last_speeds)

def get_mode():
    """Read the current mode from the mode file. Create it with default 'auto' if not found."""
    if not os.path.exists(MODE_FILE):
        try:
            with open(MODE_FILE, "w") as f:
                f.write("auto")
            logging.info(f"Mode file not found. Created {MODE_FILE} with default mode 'auto'.")
        except IOError as e:
            logging.error(f"Error creating mode file: {e}")
            return "auto" # Fallback to default
    try:
        with open(MODE_FILE, "r") as f:
            return f.read().strip().lower()
    except IOError as e:
        logging.error(f"Error reading mode file: {e}")
        return "auto" # Fallback to default

def get_manual_speed():

```

```

"""Read the manual speed from the speed file."""
try:
    with open(SPEED_FILE, "r") as f:
        speed = int(f.read().strip())
        return max(0, min(speed, 100)) # Clamp between 0 and 100
except (FileNotFoundError, ValueError):
    return None # No valid manual speed

def control_fan():
    """Main function to control the fan."""
    global last_logged_speed
    logging.info("Fan control script started.")
    try:
        while True:
            mode = get_mode()

            if mode == "manual":
                manual_speed = get_manual_speed()
                if manual_speed is not None:
                    set_fan_speed(manual_speed)
                    if last_logged_speed != manual_speed:
                        logging.info(f"Manual mode: Setting fan speed to {manual_speed}%")
                        last_logged_speed = manual_speed
                    else:
                        logging.warning("Manual mode: No valid manual speed provided, retaining
previous setting.")
                else: # Auto mode
                    temp = get_temperature()
                    if temp is not None:
                        raw_speed = calculate_fan_speed(temp)
                        smoothed_speed = smooth_speed(raw_speed)
                        set_fan_speed(smoothed_speed)

                        # Log only if the fan speed changes
                        if last_logged_speed is None or abs(smoothed_speed - last_logged_speed) >= 1:
                            logging.info(
                                f"Auto mode: CPU Temperature: {temp:.2f} °C, Setting fan speed to
{smoothed_speed:.2f}%"
                            )
                            last_logged_speed = smoothed_speed

                    time.sleep(10) # Adjust every 10 seconds (modify as needed)
            except KeyboardInterrupt:
                logging.info("Fan control script stopped.")
                set_fan_speed(MIN_SPEED if ALWAYS_ON else 0) # Leave at MIN_SPEED if
ALWAYS_ON is enabled
                pi.stop() # Clean up pigpio

```

```
if __name__ == "__main__":  
    control_fan()
```