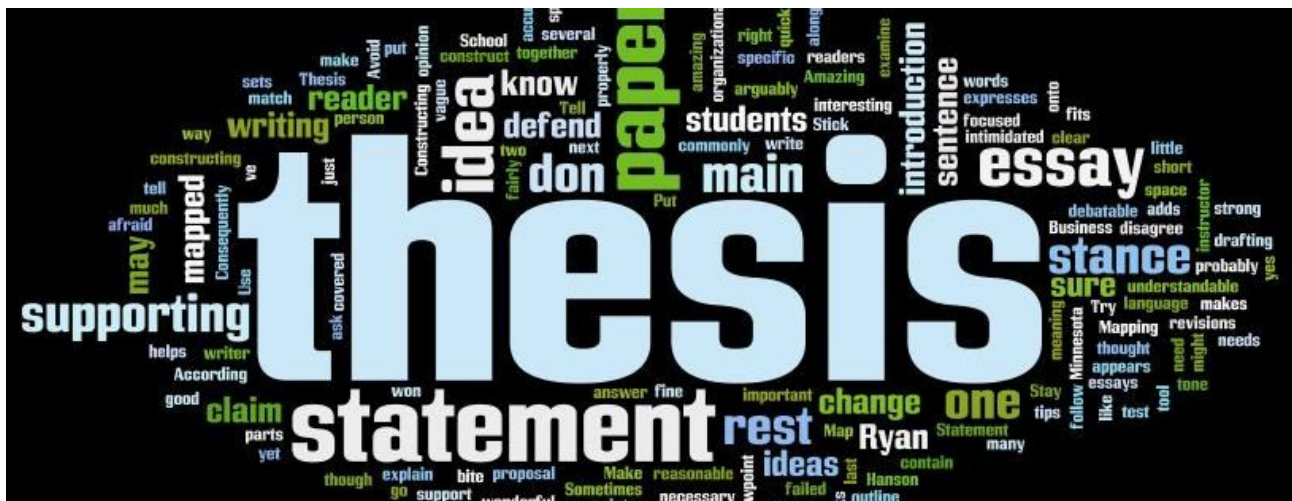




ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

Εξερεύνηση Στατικών Γεννητριών για την Ανάπτυξη Σύγχρονων Ιστοσελίδων

ΜΙΑ ΟΛΟΚΛΗΡΩΜΕΝΗ ΠΡΟΣΕΓΓΙΣΗ ΣΕ ΕΙΚΟΝΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ
ΕΦΑΡΜΟΓΩΝ.

Ευαγγελία Κατσαντώνη
(3121300)

Μάθημα: Web

Διδάσκων: Όμηρος Ιατρέλης

ΛΑΡΙΣΑ, ΜΑΪΟΣ 2025

Copyright © 2025 Ευαγγελία Κατσαντώνη



Περίληψη

Η παρούσα πτυχιακή εργασία επικεντρώνεται στην ανάπτυξη ενός συστήματος διαχείρισης Web Documentation με χρήση στατικών γεννητριών ανάπτυξης ιστοσελίδων (Vuepress). **Στόχος είναι η δημιουργία ενός περιβάλλοντος, που θα διευκολύνει την οργάνωση, τη διαχείριση και τη δημοσίευση του περιεχομένου μιας πτυχιακής εργασίας σε περιβάλλον Linux.** Η επίτευξη του στόχου πραγματοποιείται με την εφαρμογή τεχνολογιών εικονικοποίησης, συνθέτοντας ένα αποτελεσματικό και ευέλικτο σύστημα, για την εγκατάσταση και τη διαχείριση των απαραίτητων εφαρμογών.

Το κύριο Virtual Machine δημιουργείται με τη χρήση του Oracle VirtualBox, επιτρέποντας την εκτέλεση του συστήματος σε όλα τα δημοφιλή λειτουργικά (Windows, macOS, Linux). Εντός αυτής της εικονικής μηχανής εγκαθίστανται, με τη χρήση της τεχνολογίας LXD, τρία containers κάθε ένα με διαφορετικό ρόλο. Το πρώτο container φιλοξενεί το Vuepress για την ανάπτυξη του ιστότοπου documentation, το δεύτερο δημιουργεί αντίγραφα ασφαλείας για τα δεδομένα και το τρίτο δημοσιεύει τον ιστότοπο με το περιεχόμενο της πτυχιακής εργασίας. Συγχρόνως με την κύρια εικονική μηχανή παρέχεται ακόμη μία, η οποία λειτουργεί ως εργαστήριο για τον αναγνώστη, επιτρέποντας την εφαρμογή της πτυχιακής εργασίας.

Στο σύστημα ενσωματώνονται αυτοματισμοί εντολών για την ανάπτυξη και τη δημοσίευση του περιεχομένου του ιστότοπου, καθώς και cronjobs για τον αυτοματισμό των αντιγράφων ασφαλείας. Παράλληλα παρουσιάζεται οπτικό υλικό για να εξηγηθούν οι τεχνικές διαδικασίες. Πριν από την εγκατάσταση του συστήματος τοπικά, ο χρήστης έχει πρόσβαση σε ένα διαδικτυακό ιστότοπο με όλο το απαραίτητο υλικό. Όλες οι ενέργειες των αναγνωστών γίνονται μέσω web access, χωρίς να απαιτείται καμία εφαρμογή (software) πέραν του Oracle VirtualBox και χωρίς να απαιτούνται τεχνικές γνώσεις. **Το Documentation περιγράφει το αντικείμενο της εργασίας αναλυτικά σε ενότητες.**

Λέξεις κλειδιά

Static Site Generator, Documentation, Linux,
Virtualization



Περιεχόμενα

Περιεχόμενα

Περίληψη

Περιεχόμενα

Εισαγωγή

Κυρίως Μέρος

- 01 Linux

[Εισαγωγή](#) | [Ενότητα 01 Linux](#) [Επισκόπηση Linux](#) [Basic sysadmin commands](#)

[Linux Permissions](#) [Shell scripting](#) [Server Installation](#) [Lxd-Lxc commands](#)

- 02 Virtualization

[Virtualization](#) | [Ιστορία](#) [Ορισμός - Περιγραφή](#) [Software](#)

- 03 Containerization

[Containerization](#) | [Ιστορία](#) [Ορισμός - Περιγραφή](#) [Software](#)

[Containerization Vs Virtualization](#) [LXC](#) [LXD](#)

- 04 Thesis Virtual Machine

[Εισαγωγή](#) | [Ενότητα 04 Thesis Virtual Machine](#) [VirtualBox Installation](#)

[Εισαγωγή εικονικής μηχανής](#) [Εγκατάσταση Virtual Server](#) [VM-Srv-Settings](#)

[Server Administration](#) [Containers Administration](#) [4.1 Developpe Server](#) [4.2 Web Server](#)

[4.3 Cloud Server](#) [4.4 Lab Server](#)

- 05 Vuepress

[Εισαγωγή](#) | [Ενότητα 05 VuePress SSG](#) [Δημοφιλείς SSG](#) | [Σύγκριση](#) [SSG](#) | [Ανάλυση & Λειτουργία](#)

[Vuepress | Γενικά](#)[VuePress | Αρχιτεκτονική](#)[Vuepress | Εγκατάσταση](#)[Επεκτάσεις](#)[Frontmatter](#)[Διαμόρφωση](#)

- **06 Thesis Lab**

[Containers Administration](#)[Εισαγωγή | Ενότητα 06 Lab](#)

- **07 Backup**

[Create Backup](#)[Restore backup](#)

Παραρτήματα

- Λεξιλόγιο
- Βιβλιογραφία

X-DevArea

- Workplan
- Setup-workflow
- MyHacks

My Thesis (έντυπη μορφή)

Servers - Virtualization Technologies - Website Documentation (3121300/2024)

Copyright © 2025 Ευαγγελία Κατσαντώνη



Εισαγωγή

Σκιαγραφώντας το Ψηφιακό Θεμέλιο

Η παρούσα πτυχιακή εργασία παρουσιάζει ένα ολοκληρωμένο σύστημα διαχείρισης **web documentation**. Η τεχνολογική του υποδομή βασίζεται σε **στατικές γεννήτριες ιστοσελίδων (SSG)**, με το *VuePress* να δημιουργεί το frontend, τον *Apache Web Server* να εξυπηρετεί τον ιστότοπο και τα *Linux containers* να διαχειρίζονται τις εφαρμογές.

Στόχος του συστήματος είναι η δημιουργία ενός εύχρηστου και αποτελεσματικού περιβάλλοντος, προσφέροντας μια πρακτική λύση για την οργάνωση, τη διαχείριση και τη δημοσίευση περιεχομένου.

Το σύστημα παρέχει ολοκληρωμένη διαχείριση περιεχομένου και μία σειρά από ισχυρές λειτουργίες. Αυτές περιλαμβάνουν την *άμεση εκτύπωση* επιλεγμένων ενοτήτων, την *αυτόματη δημιουργία αντιγράφων ασφαλείας* και τη δυνατότητα *συνεργασίας* μέσω τοπικού δικτύου ή απομακρυσμένης πρόσβασης.

Στα Άδυστα του Συστήματος

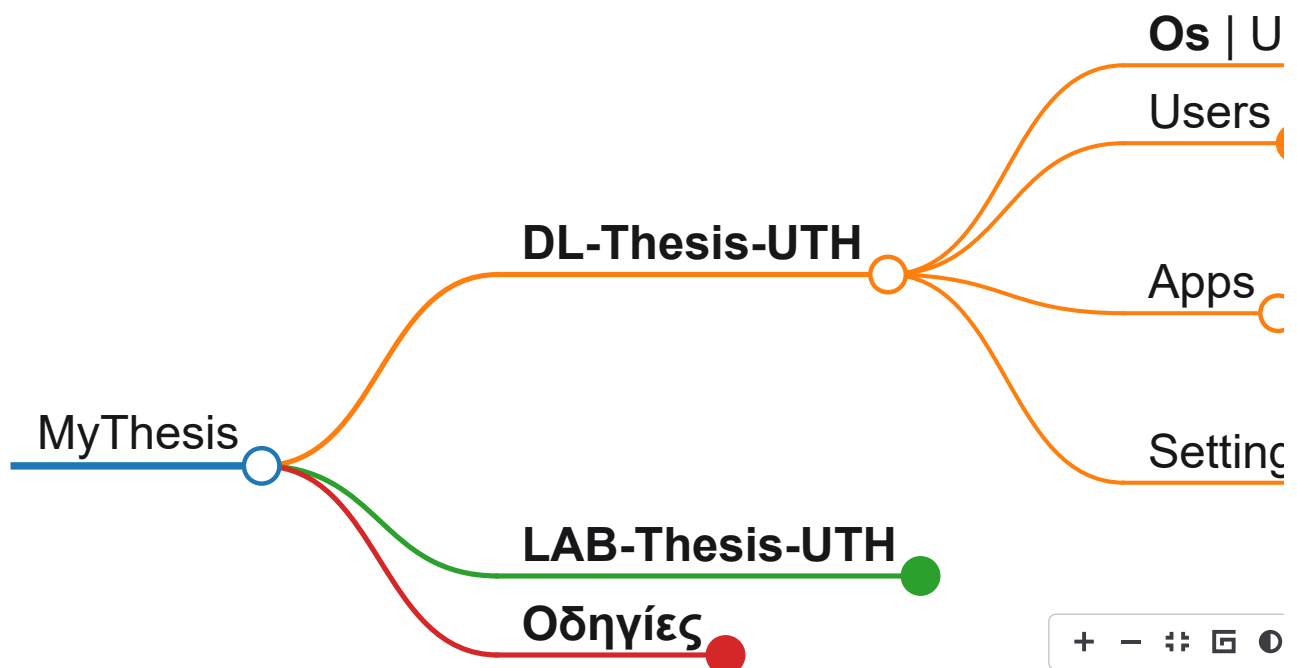
Το σύστημα λειτουργεί σε εικονική μηχανή Oracle VirtualBox, εξασφαλίζοντας πλήρη φορητότητα στα δημοφιλή λειτουργικά συστήματα. Μαζί με το χειρόγραφο, κοινοποιείται σύνδεσμος στο Cloud, όπου περιέχονται τα αρχεία της συσκευής (DL-Thesis-UTH.ova, LAB-Thesis-UTH.ova) και μια στατική σελίδα οδηγιών HTML, για την εισαγωγή αυτών στο VirtualBox (Windows, macOS και Linux). Εντός της εικονικής μηχανής (DL-Thesis-UTH.ova) είναι εγκατεστημένα τρία containers Ubuntu 24.04 LTS που εκτελούν ειδικές λειτουργίες. Ένα χρησιμοποιείται για την ανάπτυξη του ιστότοπου VuePress, ένα δεύτερο για τη δημιουργία και αποθήκευση αντιγράφων ασφαλείας και το τρίτο φιλοξενεί τον Web Server.

Το LAB-Thesis-UTH.ova είναι ένα αυτόνομο Virtual Machine, το οποίο λειτουργεί ανεξάρτητα από την εργασία. Σε αυτό τρέχει ένας Ubuntu Server 24.04 LTS, του οποίου ο ρόλος είναι να λειτουργεί ως εργαστήριο για τον αναγνώστη, επιτρέποντάς του να εφαρμόσει πλήρως την πτυχιακή εργασία σε ένα διαδραστικό περιβάλλον.

Για τη βελτιστοποιημένη διαχείριση της πλατφόρμας, ενσωματώθηκαν αυτοματισμοί εντολών (scripts και cronjobs), οι οποίοι απλοποιούν τις κρίσιμες διαδικασίες όπως η δημοσίευση και η δημιουργία-επαναφορά αντιγράφων ασφαλείας. Αυτή η προσέγγιση μειώνει την ανάγκη χειροκίνητων παρεμβάσεων, εξοικονομώντας χρόνο και ελαχιστοποιώντας τα λάθη. Επιπλέον, η αρχιτεκτονική του συστήματος εξασφαλίζει την απομόνωση των εφαρμογών, επιτρέποντας σε κάθε container να λειτουργεί ανεξάρτητα, γεγονός που διευκολύνει τη διαχείριση και την επαναφορά αποθηκευμένων στιγμιότυπων, σε περίπτωση προβλημάτων. Αξίζει να αναφερθεί ότι ακόμα και σε περίπτωση ολικής καταστροφής του container που χρησιμοποιείται για την ανάπτυξη, με μια εντολή στο command line αυτό επανέρχεται στο αρχικό που παραδόθηκε με την εργασία. Οι δυνατότητες αυτές καθιστούν το σύστημα ιδανικό για χρήστες με διαφορετικά επίπεδα τεχνικών γνώσεων και μαθητές, καθώς προσφέρει ευελιξία και ευκολία στη χρήση.

Εναρμονίζοντας το πλευρικό μενού του ιστοτόπου με τη δομή μιας πτυχιακής εργασίας, έχουν δημιουργηθεί αντίστοιχες συνδέσεις στα περιεχόμενα, στην περίληψη, στην εισαγωγή, στο κυρίως μέρος (με τα επιμέρους κεφάλαιά του), καθώς και στα παραρτήματα. Ειδικά για τη μελέτη της πτυχιακής, περιλαμβάνεται ένα αναπτυσσόμενο μενού, ομαδοποιημένο ανά container, με την ονομασία 'Admin's Bookmarks'. Αυτό περιέχει εξειδικευμένες συνδέσεις σε εργαλεία και εφαρμογές, διευκολύνοντας την πρόσβαση στη διαχείριση των πόρων του server. Ο αναγνώστης μπορεί να ακολουθήσει τις διαδικασίες μέσω της λεπτομερούς τεκμηρίωσης με οπτικό υλικό, χωρίς να χρειάζεται ειδική τεχνική γνώση ή εγκατάσταση περισσότερων εφαρμογών πέραν του Oracle VirtualBox και να πειραματιστεί στον εικονικό Server LAB-THESIS-SRV, ο οποίος σχεδιάστηκε για εργαστήριο στην εφαρμογή της μελέτης.

Γραφική δομή της πτυχιακής μου εργασίας
(επιλέξτε τις χρωματιστές κουκίδες για λεπτομερείες)



Αποκωδικοποιώντας την Αρχιτεκτονική

Στο *Κυρίως Μέρος* της εργασίας περιγράφονται ιεραρχικά και αναλυτικά σε κεφάλαια πολλές πληροφορίες, ώστε ο απλός χρήστης με βασικές γνώσεις πληροφορικής να κατανοήσει το αντικείμενο της μελέτης. Διαβάζοντας προσεκτικά αυτόν τον οδηγό, ο αναγνώστης θα αποκτήσει μια γενική γνώση για το λειτουργικό σύστημα Linux, καθώς και για τις τεχνολογίες εικονικοποίησης (virtualization , containerization), οι οποίες αποτελούν τη βάση της παρούσας μελέτης.

Η εισαγωγή στην πλατφόρμα VuePress, η οποία αξιοποιήθηκε για τη συγγραφή της εργασίας, περιλαμβάνει την εγκατάσταση και τις βασικές τεχνικές παραμετροποιήσεις. Προκειμένου να αναδειχθούν πλήρως οι δυνατότητες του θέματος, όπως η διαχείριση περιεχομένου σε μορφή blog, η οργάνωση των άρθρων σε κατηγορίες και ετικέτες, καθώς και η χρονολογική ταξινόμηση, επιλεγμένα κεφάλαια της εργασίας έχουν αναδιαμορφωθεί ή δημιουργηθεί ως άρθρα στην τεχνική βιβλιοθήκη. Συνεχίζοντας στην τεχνική υποδομή του διακομιστή, το κεφάλαιο 'Thesis Virtual Machine' προσφέρει στον αναγνώστη πρακτική εμπειρία στην συνολική ανάπτυξη του έργου, έχοντας παράλληλα τη δυνατότητα εφαρμογής στο εργαστήριο του συστήματος.

Πολύτιμος βοηθός του χρήστη είναι το κεφάλαιο 'Backup', στο οποίο αναφέρονται μέθοδοι δημιουργίας και επαναφοράς αντιγράφων ασφαλείας, προσφέροντάς του αυτοπεποίθηση σε κάθε του βήμα. Τέλος, στην ενότητα 'Παραρτήματα' περιλαμβάνονται το Λεξιλόγιο και η Βιβλιογραφία. Στο Λεξιλόγιο περιέχονται όροι που αναφέρονται στα κείμενα της εργασίας, με αυτόματες παραπομπές, ενώ στη σελίδα της Βιβλιογραφίας υπάρχουν εξωτερικοί σύνδεσμοι στις πηγές και στους οδηγούς. Αξιοσημείωτο είναι ότι διατηρούνται ορισμένες

βασικές πηγές σε αρχεία μορφότυπου PDF, τα οποία είναι τοπικά αποθηκευμένα στον server, εξασφαλίζοντας την ανάγνωση σε περιπτώσεις τροποποιήσεων των εξωτερικών ιστοσελίδων.

Επίλογος

Αυτή η εργασία φιλοδοξεί να προσφέρει μια ολοκληρωμένη και προσβάσιμη λύση για τη διαχείριση του web documentation, ενισχύοντας την εμπειρία του χρήστη και διευκολύνοντας τη διαδικασία ανάπτυξης και διαχείρισης περιεχομένου.

Κυρίως Μέρος

 [Ευαγγελία Κατσιαντώνη](#)  Λιγότερο από 1 λεπτό

Περιεχόμενα

01 Linux

- Εισαγωγή | Ενότητα 01 Linux
- Επισκόπηση Linux
- Basic sysadmin commands
- Linux Permissions
- Shell scripting
- Server Installation
- Lxd-Lxc commands

02 Virtualization

- Virtualization | Ιστορία
- Ορισμός - Περιγραφή
- Software

03 Containerization

- Containerization | Ιστορία
- Ορισμός - Περιγραφή
- Software
- Containerization Vs Virtualization
- LXC
- LXD

04 Thesis Virtual Machine

- 4.1 Developpe Server
[CT-001-DEV](#)
- 4.2 Web Server
[CT-003-WEB](#)
- 4.3 Cloud Server

CT-002-FILE

- 4.4 Lab Server

CT-003-LAB

- Εισαγωγή | Ενότητα 04 Thesis Virtual Machine
- VirtualBox Installation
- Εισαγωγή εικονικής μηχανής
- Εγκατάσταση Virtual Server
- VM-Srv-Settings
- Server Administration
- Containers Administration

05 Vuepress

- Εισαγωγή | Ενότητα 05 VuePress SSG
- Δημοφιλείς SSG | Σύγκριση
- SSG | Ανάλυση & Λειτουργία
- Vuepress | Γενικά
- VuePress | Αρχιτεκτονική
- Vuepress | Εγκατάσταση
- Επεκτάσεις
- Frontmatter
- Διαμόρφωση

06 Thesis Lab

- Containers Administration
- Εισαγωγή | Ενότητα 06 Lab

07 Backup

- Create Backup
- Restore backup



Εισαγωγή | Ενότητα 01 Linux

👤 Ευαγγελία Κατσιαντώνη 📅 5/21/25 🏢 Linux 💎 Εισαγωγή ⌚ Λιγότερο από 1 λεπτό

Σε αυτή την ενότητα, θα αναλύσουμε ορισμένες από τις βασικές αρχές και πρακτικές του Linux, εστιάζοντας ειδικότερα στο Ubuntu. Ο στόχος είναι να αποκτήσετε τις απαραίτητες γνώσεις και δεξιότητες για τον αποδοτικό χειρισμό του λειτουργικού συστήματος, οι οποίες είναι θεμελιώδεις τόσο για την πλήρη κατανόηση της παρούσας εργασίας όσο και για τον απρόσκοπτο πειραματισμό σας στο εργαστηριακό περιβάλλον.

Για την επίτευξη αυτού του στόχου, θα καλυφθούν θέματα όπως οι βασικές εντολές διαχείρισης συστήματος (sysadmin commands) και τα δικαιώματα αρχείων (Linux Permissions). Επιπλέον, θα εξεταστεί η δύναμη του shell scripting και οι ειδικές εντολές για τα εικονικά περιβάλλοντα LXD/LXC. Κάθε υποενότητα έχει σχεδιαστεί για να σας εξοπλίσει με τα απαραίτητα εργαλεία, ώστε με την ολοκλήρωση αυτής της ενότητας, να έχετε μια στέρεη βάση για να προχωρήσετε στις πιο εξειδικευμένες ενότητες της εργασίας και να εφαρμόσετε τις γνώσεις σας στην πράξη.

Επισκόπηση Linux

👤 Ευαγγελία Κατσιαντώνη 📅 5/21/25 🖥️ Linux 🔍 Επισκόπηση ⌚ Περίπου 9 λεπτά

Εισαγωγή

Στις παρακάτω γραμμές γίνεται μια συνοπτική περιγραφή του **Linux**, ενός λειτουργικού συστήματος ανοικτού κώδικα που έχει ανατρέψει τα δεδομένα της πληροφορικής. Από την αρχική του εμφάνιση μέχρι και σήμερα, το Linux έχει εξελιχθεί σε μια πολυμορφική οικογένεια διανομών, καθιστώντας το μία ισχυρή και ευέλικτη λύση για μια πληθώρα εφαρμογών.

Ταυτότητα του Linux

Το Linux είναι ένα ελεύθερο και ανοιχτού κώδικα **λειτουργικό σύστημα με παγκόσμια δημοτικότητα**. Δημιουργήθηκε το 1991 από τον **Linus Torvalds**, όταν τότε ως φοιτητής Πληροφορικής στο Πανεπιστήμιο του Ελσίνκι της Φινλανδίας, δημιούργησε τον πυρήνα (kernel^[1]) του λειτουργικού συστήματος. Η πρωτοποριακή απόφαση να διατεθεί ο κώδικας υπό την άδεια GNU General Public License (GPL)^[2] έθεσε τα θεμέλια για το open-source κίνημα, δημιουργώντας ένα μοναδικό οικοσύστημα αμοιβαίας συνεργασίας και βελτίωσης. Έκτοτε έχει εξελιχθεί σε μια ισχυρή πλατφόρμα που χρησιμοποιείται σε υπολογιστές, διακομιστές, κινητές συσκευές και πολλά άλλα. Σε σύγκριση με άλλα λειτουργικά συστήματα, όπως τα Windows και το macOS, το Linux προσφέρει μεγαλύτερη ευελιξία και προσαρμοστικότητα, καθώς και μια ενεργή κοινότητα υποστήριξης.

Διανομές (Distributions)

Η ανοικτή φύση του Linux οδήγησε στη δημιουργία πολλών διανομών, γνωστών ως distros, καθεμία με τα δικά της χαρακτηριστικά και προσανατολισμό. Οι διανομές αποτελούν βασική έννοια του Linux, παρέχοντας διαφορετικές εκδόσεις του λειτουργικού συστήματος, συχνά με διαφορετικά λογισμικά και εργαλεία. Δημοφιλείς διανομές περιλαμβάνουν το **Ubuntu**, το **Fedora**, το **Debian** και το **Arch Linux**, καθεμία με ιδιαίτερα χαρακτηριστικά και προσαρμογές. Η επιλογή της κατάλληλης διανομής εξαρτάται από τις ανάγκες και τις προτιμήσεις του χρήστη.

Γραφικό περιβάλλον

Η ελευθερία που προσφέρει το Linux εξάπτει τη δημιουργικότητα και την καινοτομία στην ανάπτυξη Desktop Environments (DE).

Κάθε ένα από αυτά προσφέρει μία μοναδική εμπειρία χρήστη, καθιστώντας το Linux μία ευέλικτη λύση για μια πληθώρα αναγκών.

- **Ορισμός:** *Τι είναι όμως τα Desktop Environments που προσδίδουν ιδιαίτερα χαρακτηριστικά και δυναμική στις διανομές Linux;*
Ο όρος Desktop Environments (Περιβάλλοντα Επιφάνειας Εργασίας), **Γραφικό περιβάλλον** εν Ελληνιστί, είναι σύνολα λογισμικού που παρέχουν μια γραφική διεπαφή χρήστη (GUI) για την αλληλεπίδραση με το λειτουργικό σύστημα. Αυτά τα περιβάλλοντα περιλαμβάνουν όλα τα απαραίτητα στοιχεία και την οπτική εικόνα που χρειάζεται ο χρήστης για να αλληλεπιδράσει με τον υπολογιστή του.
- **Βασικά Στοιχεία των Desktop Environments**
 - **Γραφικά Στοιχεία**
Περιλαμβάνουν τα παράθυρα, το μενού, τα κουμπιά, τα εικονίδια και άλλες γραφικές αναπαραστάσεις που διευκολύνουν την πλοήγηση και τη χρήση του υπολογιστή.
 - **Διαχείριση Παραθύρων**
Παρέχουν τη δυνατότητα να ανοίγουν, να κλείνουν, να ελαχιστοποιούν και να μεγιστοποιούν τα παράθυρα των εφαρμογών.
 - **Εργαλεία και Εφαρμογές**
Συχνά περιλαμβάνουν προεγκατεστημένες εφαρμογές, όπως διαχειριστές αρχείων, τερματικά, προγράμματα περιήγησης και εργαλεία ρυθμίσεων.
 - **Προσαρμογή**
Πολλά περιβάλλοντα επιφάνειας εργασίας επιτρέπουν στους χρήστες να προσαρμόζουν την εμφάνιση και τη λειτουργικότητα, όπως τα θέματα, τα εικονίδια και τις ρυθμίσεις του πάνελ.
- **Δημοφιλή Desktop Environments**
 - **GNOME**
Ένα σύγχρονο και διαισθητικό περιβάλλον, γνωστό για την ευκολία χρήσης και την ομαλή λειτουργία. Προσφέρει ένα συνοπτικό και εστιασμένο σε λειτουργίες GUI, ιδανικό για αρχάριους και όσους θέλουν ένα άψογο οπτικό τοπίο.
 - **KDE Plasma**
Προσφέρει υψηλή προσαρμοστικότητα και ένα πλούσιο σύνολο εργαλείων για τους προχωρημένους χρήστες. Χαρακτηρίζεται από μία πιο επεκτατική και flexible

προσέγγιση, που επιτρέπει να αξιοποιηθούν όλες οι δυνατότητες του Linux μέσα στο πλαίσιο του desktop environment.

- **XFCE**

Ένα ελαφρύ και γρήγορο περιβάλλον, ιδανικό για συσκευές με περιορισμένους πόρους ή όσους θέλουν μία ταχύτερη και απλούστερη εμπειρία. Εξαιρετική λύση για παλαιότερα συστήματα ή όσους αγαπούν μία πιο minimal εμπειρία χρήστη.

- **MATE**

Ένα πιο παραδοσιακό περιβάλλον με γνώμονα την ευκολία χρήσης, κατάλληλο για όσους είναι εξοικειωμένοι με παλαιότερες εκδόσεις Linux. Προσφέρει μια οικεία εμπειρία για όσους επικεντρώνονται στην απλότητα και την αξιοπιστία.

- **Cinnamon**

Χτισμένο πάνω στη βάση του GNOME Shell, το Cinnamon προσφέρει μία πιο παραδοσιακή εμπειρία desktop με panel και menu που θυμίζουν το παλαιότερο GNOME. Είναι ιδανικό για όσους θέλουν μια πιο οικεία εμπειρία χωρίς να χρειάζεται να μετακινηθούν σε ένα πλήρως νέο GUI.

- **LXQt**

Ένα ελαφρύ και γρήγορο περιβάλλον desktop, ιδανικό για συσκευές με περιορισμένους πόρους ή όσους θέλουν ένα εύχρηστο και απλό GUI. Παρόμοιο με το XFCE, αλλά με μια πιο σύγχρονη εμφάνιση και πιο minimal σχεδιασμό.

Tips

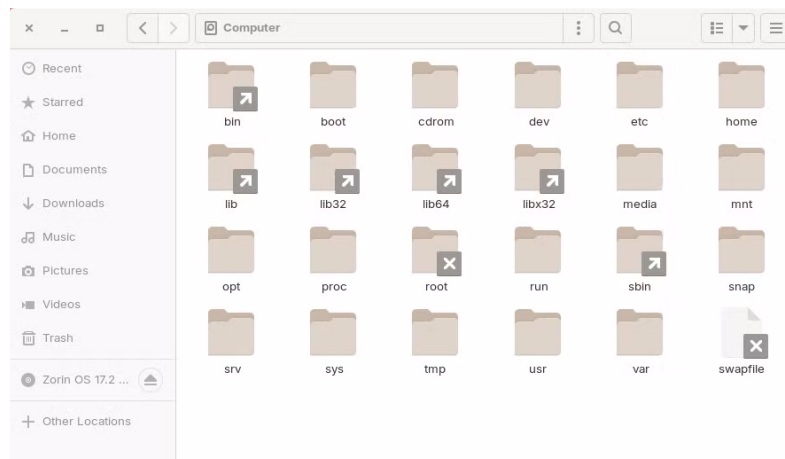
Πολλές διανομές Linux επιτρέπουν την εγκατάσταση περισσότερων από ενός desktop environments, δίνοντας στους χρήστες την ευελιξία να επιλέξουν το περιβάλλον που ταιριάζει στις προτιμήσεις τους, ακόμα και μετά την αρχική εγκατάσταση της διανομής

Η επιλογή του κατάλληλου desktop environment είναι θέμα προσωπικής προτίμησης και ανάγκης.

Σύστημα Αρχείων

Το σύστημα αρχείων είναι μια θεμελιώδης πτυχή του Linux, καθώς καθορίζει τον τρόπο με τον οποίο οργανώνονται και αποθηκεύονται τα δεδομένα. Το **ext4** (*fourth extended filesystem*) είναι ένα από τα πιο δημοφιλή και ευρέως χρησιμοποιούμενα συστήματα αρχείων στο Linux. Όλα τα αρχεία και οι φάκελοι οργανώνονται σε μια ιεραρχική δομή, γνωστή ως Filesystem Hierarchy Standard (FHS) ή δέντρο αρχείων, με τη ρίζα (/) στην

κορυφή. Αυτή η δομή επιτρέπει στους χρήστες να πλοηγούνται εύκολα στα δεδομένα τους.



Εικόνα 1.1: Ιεραρχική Δομή Συστήματος Αρχείων Linux

1. / (Root Directory):

- Ο ριζικός κατάλογος είναι η κορυφή της ιεραρχίας του συστήματος αρχείων. Όλοι οι άλλοι κατάλογοι και αρχεία βρίσκονται κάτω από αυτόν.

2. /bin:

- Περιέχει βασικά εκτελέσιμα αρχεία (εντολές) που απαιτούνται για τη λειτουργία του συστήματος, όπως οι εντολές `ls`, `cp`, `mv`, κ.λπ. Αυτές οι εντολές είναι διαθέσιμες σε όλους τους χρήστες.

3. /boot:

- Περιέχει τα αρχεία που απαιτούνται για την εκκίνηση του συστήματος, όπως ο πυρήνας (kernel) και οι ρυθμίσεις εκκίνησης.

4. /dev:

- Περιέχει ειδικά αρχεία συσκευών (device files) που αντιπροσωπεύουν τις συσκευές του συστήματος, όπως σκληρούς δίσκους, USB, κ.λπ.

5. /etc:

- Περιέχει αρχεία ρυθμίσεων (configuration files) για το σύστημα και τις εφαρμογές. Αυτά τα αρχεία είναι συνήθως κείμενα και περιέχουν ρυθμίσεις για διάφορες υπηρεσίες.

6. /home:

- Περιέχει τους προσωπικούς καταλόγους των χρηστών. Κάθε χρήστης έχει τον δικό

του υποκατάλογου εδώ, π.χ. `/home/username` .

7. `/lib`:

- Περιέχει τις βιβλιοθήκες (libraries) που απαιτούνται από τα εκτελέσιμα αρχεία στο `/bin` και `/sbin` .

8. `/media`:

- Χρησιμοποιείται για την τοποθέτηση (mounting) εξωτερικών συσκευών αποθήκευσης, όπως USB drives και CD/DVD.

9. `/mnt`:

- Χρησιμοποιείται για προσωρινή τοποθέτηση (mounting) συσκευών αποθήκευσης από τον διαχειριστή του συστήματος.

0. `/opt`:

- Χρησιμοποιείται για την εγκατάσταση προαιρετικών εφαρμογών και πακέτων λογισμικού που δεν περιλαμβάνονται στο βασικό σύστημα.

1. `/proc`:

- Ένας εικονικός κατάλογος που περιέχει πληροφορίες σχετικά με τη λειτουργία του συστήματος και τις διεργασίες. Τα αρχεία εδώ δεν είναι πραγματικά αρχεία στο δίσκο, αλλά παρέχουν πληροφορίες σε πραγματικό χρόνο.

2. `/root`:

- Ο προσωπικός κατάλογος του χρήστη root (superuser). Είναι διαφορετικός από το `/home/root` .

3. `/run`:

- Περιέχει δεδομένα που σχετίζονται με την τρέχουσα εκτέλεση του συστήματος, όπως PID files και sockets.

4. `/srv`:

- Χρησιμοποιείται για δεδομένα που παρέχονται από υπηρεσίες που εκτελούνται στο σύστημα, όπως ιστοσελίδες ή FTP.

5. `/sys`:

- Ένας εικονικός κατάλογος που παρέχει πληροφορίες σχετικά με το σύστημα και τις

συσκευές του, παρόμοιος με το `/proc` .

6. `/tmp`:

- Χρησιμοποιείται για προσωρινά αρχεία που μπορεί να διαγραφούν κατά την επανεκκίνηση του συστήματος.

7. `/usr`:

- Περιέχει δεδομένα που είναι κοινά για όλους τους χρήστες, όπως εφαρμογές, βιβλιοθήκες και έγγραφα. Υποκατάλογοι περιλαμβάνουν:
- `/usr/bin`: Εκτελέσιμα αρχεία για εφαρμογές.
- `/usr/lib`: Βιβλιοθήκες για εφαρμογές.
- `/usr/share`: Κοινά δεδομένα, όπως έγγραφα και εικόνες.

8. `/var`:

- Περιέχει μεταβλητά δεδομένα, όπως αρχεία καταγραφής (logs), βάσεις δεδομένων και αρχεία που συχνά αλλάζουν μέγεθος.

Εγκατάσταση

Η εγκατάσταση του Linux είναι μια διαδικασία που μπορεί να γίνει εύκολα, ακολουθώντας μερικά βήματα. Οι χρήστες μπορούν να δοκιμάσουν μια διανομή μέσω ενός Live USB ή CD πριν από την εγκατάσταση, ώστε να δουν αν τους ταιριάζει. Επίσης, είναι δυνατή η εγκατάσταση του Linux σε διπλή εκκίνηση (dual boot) με Windows, επιτρέποντας στους χρήστες να επιλέγουν ποιο λειτουργικό σύστημα θα χρησιμοποιήσουν κατά την εκκίνηση του υπολογιστή.

Βασικές εντολές του τερματικού (Terminal)

Το τερματικό είναι ένα ισχυρό εργαλείο στο Linux που επιτρέπει στους χρήστες να αλληλεπιδρούν με το σύστημα μέσω εντολών. Ορισμένες βασικές εντολές περιλαμβάνουν το `ls` για την εμφάνιση περιεχομένων φακέλων, το `cd` για την πλοήγηση στους φακέλους, και το `cp`, `mv`, `rm`, και `mkdir` για τη διαχείριση αρχείων και φακέλων. Η εντολή `man` μπορεί να χρησιμοποιηθεί για να αποκτήσουν βοήθεια σχετικά με άλλες εντολές. Περισσότερα στο άρθρο [Basic sysadmin commands](#)

Διαχείριση πακέτων

Η διαχείριση πακέτων είναι μια σημαντική πτυχή του Linux, καθώς επιτρέπει στους χρήστες να εγκαθιστούν και να απεγκαθιστούν λογισμικό εύκολα. Κάθε διανομή έχει το δικό της σύστημα διαχείρισης πακέτων, όπως το `apt` για το Debian/Ubuntu και το `dnf` για το Fedora. Οι χρήστες μπορούν επίσης να ενημερώνουν το σύστημα τους με απλές εντολές.

Ασφάλεια και δικαιώματα

Η ασφάλεια είναι κρίσιμη στο Linux, και οι χρήστες πρέπει να κατανοήσουν τα δικαιώματα χρηστών και ομάδων. Η χρήση του `sudo` επιτρέπει στους χρήστες να εκτελούν εντολές με δικαιώματα διαχειριστή, ενώ η κατανόηση του χρήστη **root** είναι επίσης σημαντική. Οι βασικές πρακτικές ασφαλείας περιλαμβάνουν την τακτική ενημέρωση του συστήματος και την προσεκτική διαχείριση των δικαιωμάτων. Μια αναλυτικότερη προσέγγιση αναφορικά με τα δικαιώματα στο Linux περιγράφεται παρακάτω στο κεφάλαιο [Linux Permissions](#) .

Κοινότητα και υποστήριξη

Η κοινότητα του Linux είναι μια από τις πιο δυναμικές και υποστηρικτικές. Υπάρχουν πολλές πηγές για βοήθεια, όπως φόρουμ, IRC και ιστοσελίδες όπως το Reddit. Ακολουθούν τρεις από τις πιο γνωστές πηγές βοήθειας για την κοινότητα του Linux σε κάθε κατηγορία, μαζί με τα αντίστοιχα links:

Φόρουμ

1. Γενικά για Linux:

- Επίσημο φόρουμ: [LinuxQuestions.org](https://linuxquestions.org)

2. Ubuntu:

- Επίσημο φόρουμ: [Ubuntu Forums](https://ubuntuforums.org)
- Κοινότητα: [Ask Ubuntu](https://askubuntu.com)

3. Arch Linux:

- Επίσημο φόρουμ: [Arch Linux Forums](https://bbs.archlinux.org)
- Κοινότητα: [Arch Wiki](https://wiki.archlinux.org)

IRC

Για την είσοδό σας στο Chat ακολουθήστε το σύνδεσμο <https://web.libera.chat/> στο

πεδίο **Ψευδώνυμο** πληκτρολογήστε ένα ψευδώνυμο της επιλογής σας ή αφήστε την προεπιλογή και στο πεδίο **Κανάλι** ένα από τα #ubuntu, #archlinux, #linux, για να εισέλθετε αυτόματα χωρίς κωδικό, στο κανάλι της επιλογής σας. Το IRC (Internet Relay Chat) είναι ένα παλαιότερο, αλλά ακόμα ενεργό, σύστημα ανταλλαγής μηνυμάτων σε πραγματικό χρόνο, ιδανικό για άμεση επικοινωνία και υποστήριξη από την κοινότητα.

Reddit

1. Linux

<https://www.reddit.com/r/linux/>

2. Ubuntu

<https://www.reddit.com/r/Ubuntu/>

3. Arch

<https://www.reddit.com/r/archlinux/>

Basic sysadmin commands

👤 Ευαγγελία Κατσιαντώνη 📅 8/12/24 🖥️ Linux 🛠️ commands, cli ⌚ Περίπου 17 λεπτά

Εισαγωγή

Ένας διαχειριστής συστημάτων Linux Debian πρέπει να έχει καλή γνώση μιας σειράς βασικών κατηγοριών εντολών που είναι κρίσιμες για την αποτελεσματική διαχείριση και συντήρηση του συστήματος.

- Διαχείριση Χρηστών και Ομάδων: Αυτές οι εντολές επιτρέπουν τη ρύθμιση δικαιωμάτων πρόσβασης και την ασφάλεια του συστήματος. Οι διαχειριστές μπορούν να δημιουργούν, να τροποποιούν και να διαγράφουν χρήστες και ομάδες, διασφαλίζοντας ότι οι χρήστες έχουν τα κατάλληλα δικαιώματα για την εκτέλεση των εργασιών τους.
- Διαχείριση Πακέτων: Οι εντολές για τη διαχείριση πακέτων είναι απαραίτητες για την εγκατάσταση, την ενημέρωση και την αφαίρεση λογισμικού. Αυτές οι εντολές διασφαλίζουν ότι το σύστημα παραμένει ενημερωμένο και ασφαλές, επιτρέποντας στους διαχειριστές να διαχειρίζονται τις εξαρτήσεις και τις ενημερώσεις λογισμικού με ευκολία.
- Παρακολούθηση και Διαχείριση Πόρων: Οι εντολές που σχετίζονται με την παρακολούθηση και τη διαχείριση πόρων, όπως η μνήμη και η CPU, είναι σημαντικές για τη βελτιστοποίηση της απόδοσης του συστήματος. Οι διαχειριστές μπορούν να παρακολουθούν τη χρήση πόρων και να εντοπίζουν πιθανά προβλήματα απόδοσης.
- Διαχείριση Αρχείων και Φακέλων: Οι εντολές για τη διαχείριση αρχείων και φακέλων είναι θεμελιώδεις για την οργάνωση και την πρόσβαση στα δεδομένα. Οι διαχειριστές χρησιμοποιούν αυτές τις εντολές για να δημιουργούν, να αντιγράφουν, να μετακινούν και να διαγράφουν αρχεία και φακέλους, διασφαλίζοντας ότι η δομή των δεδομένων είναι οργανωμένη και προσβάσιμη.

Η καλή γνώση αυτών των κατηγοριών εντολών επιτρέπει στους διαχειριστές να διαχειρίζονται αποτελεσματικά το σύστημα και να διασφαλίζουν τη σταθερότητα και την ασφάλεια του περιβάλλοντος εργασίας.

Χρήστες και Ομάδες

Ο πίνακας που ακολουθεί παρέχει μια συνοπτική επισκόπηση των βασικών εντολών για την προβολή και διαχείριση χρηστών σε ένα σύστημα Debian Linux. Οι εντολές αυτές

είναι απαραίτητες για τους διαχειριστές συστημάτων, καθώς επιτρέπουν τη δημιουργία, την τροποποίηση, τη διαγραφή χρηστών, όπως επίσης την προβολή πληροφοριών σχετικά με τα δικαιώματα και τις ομάδες τους. Κάθε εντολή συνοδεύεται από μια περιγραφή της λειτουργίας της, τα ορίσματα που απαιτούνται και μια ανάλυση που εξηγεί τη διαδικασία και τη σημασία της. Αυτές οι εντολές είναι θεμελιώδεις για τη διαχείριση της ασφάλειας και της πρόσβασης σε πόρους του συστήματος, διασφαλίζοντας ότι οι χρήστες έχουν τα κατάλληλα δικαιώματα και πρόσβαση.

Διαχείριση Χρηστών

Πίνακας Εντολών

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
<code>adduser <username></code>	Δημιουργεί νέο χρήστη με το όνομα <code><username></code> .	<code><username></code> : Το όνομα του νέου χρήστη.	Δημιουργεί έναν νέο χρήστη στο σύστημα. Κατά τη διαδικασία, ζητάει πληροφορίες όπως κωδικό πρόσβασης και πλήρες όνομα.
<code>deluser <username></code>	Διαγράφει τον χρήστη με το όνομα <code><username></code> .	<code><username></code> : Το όνομα του χρήστη προς διαγραφή.	Διαγράφει τον χρήστη και, αν ζητηθεί, μπορεί να διαγράψει και τον προσωπικό του φάκελο. Χρειάζεται προσοχή, καθώς η διαγραφή είναι μόνιμη.
<code>usermod - aG <group> <username></code>	Προσθέτει τον χρήστη <code><username></code> σε μια ομάδα <code><group></code> .	<code>-aG</code> : Προσθήκη σε ομάδα χωρίς να αφαιρεθεί από άλλες.	Χρησιμοποιείται για να προσθέσει έναν χρήστη σε μια ή περισσότερες ομάδες. Το <code>-a</code> διασφαλίζει ότι ο χρήστης δεν θα αφαιρεθεί από άλλες ομάδες.
<code>deluser <username> <group></code>	Διαγράφει τον χρήστη <code><username></code> από την ομάδα <code><group></code> .	<code><username></code> : Το όνομα του χρήστη, <code><group></code> : Το όνομα της ομάδας.	Αφαιρεί τον χρήστη από την καθορισμένη ομάδα. Χρήσιμο για τη διαχείριση των δικαιωμάτων πρόσβασης.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
passwd <username>	Αλλάζει τον κωδικό πρόσβασης του χρήστη <username> .	<username> : Το όνομα του χρήστη του οποίου ο κωδικός θα αλλάξει.	Επιτρέπει την αλλαγή του κωδικού πρόσβασης ενός χρήστη. Ο διαχειριστής μπορεί να αλλάξει τον κωδικό πρόσβασης οποιουδήποτε χρήστη.
id <username>	Εμφανίζει πληροφορίες για τον χρήστη <username> , όπως UID και GID.	<username> : Το όνομα του χρήστη.	Εμφανίζει το UID (User ID), το GID (Group ID) και τις ομάδες στις οποίες ανήκει ο χρήστης. Χρήσιμο για την κατανόηση των δικαιωμάτων του χρήστη.
getent passwd	Εμφανίζει τη λίστα όλων των χρηστών του συστήματος.	-	Εμφανίζει τις πληροφορίες των χρηστών όπως στο /etc/passwd , αλλά μπορεί να περιλαμβάνει και χρήστες από άλλες πηγές (π.χ. LDAP).
groups <username>	Εμφανίζει τις ομάδες στις οποίες ανήκει ο χρήστης <username> .	<username> : Το όνομα του χρήστη.	Εμφανίζει όλες τις ομάδες στις οποίες ανήκει ο χρήστης, διευκολύνοντας την κατανόηση των δικαιωμάτων και των ρόλων του.
cat /etc/ passwd	Εμφανίζει το περιεχόμενο του αρχείου /etc/passwd .	-	Διαβάζει και εμφανίζει το αρχείο /etc/passwd , το οποίο περιέχει πληροφορίες για τους χρήστες του συστήματος.
chage -l <username>	Εμφανίζει πληροφορίες σχετικά με την πολιτική κωδικών πρόσβασης του χρήστη.	<username> : Το όνομα του χρήστη.	Παρέχει λεπτομέρειες σχετικά με την πολιτική κωδικών πρόσβασης, όπως η ημερομηνία τελευταίας αλλαγής κωδικού και οι απαιτήσεις για αλλαγή.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
who	Εμφανίζει τους χρήστες που είναι συνδεδεμένοι στο σύστημα.	-	Εμφανίζει μια λίστα με τους χρήστες που είναι αυτή τη στιγμή συνδεδεμένοι στο σύστημα, μαζί με τις λεπτομέρειες της σύνδεσής τους.
last	Εμφανίζει την ιστορία των τελευταίων συνδέσεων χρηστών.	-	Παρέχει μια λίστα με τις τελευταίες συνδέσεις χρηστών, συμπεριλαμβανομένων των ημερομηνιών και των ωρών σύνδεσης και αποσύνδεσης.

Διαχείριση Ομάδων

Πίνακας Εντολών

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
groupadd <group>	Δημιουργεί μια νέα ομάδα με το όνομα <group> .	<group> : Το όνομα της νέας ομάδας.	Δημιουργεί μια νέα ομάδα στο σύστημα, επιτρέποντας τη διαχείριση χρηστών και δικαιωμάτων.
groupdel <group>	Διαγράφει την ομάδα με το όνομα <group> .	<group> : Το όνομα της ομάδας προς διαγραφή.	Διαγράφει την ομάδα από το σύστημα. Χρειάζεται προσοχή, καθώς η διαγραφή μπορεί να επηρεάσει τους χρήστες που ανήκουν σε αυτήν.
getent group	Εμφανίζει τη λίστα όλων των ομάδων του συστήματος.	-	Εμφανίζει τις πληροφορίες των ομάδων όπως στο /etc/group , παρέχοντας μια συνοπτική εικόνα των διαθέσιμων ομάδων.

▼ Παραδείγματα Εντολών

- Δημιουργία νέου χρήστη:

```
1 sudo adduser john sh
```

- Διαγραφή χρήστη:

```
1 sudo deluser john sh
```

- Προσθήκη χρήστη σε ομάδα:

```
1 sudo usermod -aG sudo john sh
```

- Αφαίρεση χρήστη από ομάδα:

```
1 sudo deluser john sudo sh
```

- Αλλαγή κωδικού πρόσβασης:

```
1 sudo passwd john sh
```

- Εμφάνιση πληροφοριών χρήστη:

```
1 id john sh
```

- Λίστα όλων των χρηστών:

```
1 getent passwd sh
```

- Ομάδες χρήστη:

```
1 groups john sh
```

- Πληροφορίες πολιτικής κωδικού:

```
1 chage -l john sh
```

- Δημιουργία νέας ομάδας:

```
1 sudo groupadd developers sh
```

- Διαγραφή ομάδας:

1

```
sudo groupdel developers
```

sh

- Λίστα όλων των ομάδων:

1

```
getent group
```

sh

- Εμφάνιση ονόματος χρήστη, UID και πλήρους ονόματος:

1

```
cat /etc/passwd | cut -d: -f1,3,5
```

sh

Σημείωση: Χρησιμοποιούμε τον `|` (pipe) αντί του `&&` γιατί θέλουμε την έξοδο της `cat` να γίνει είσοδος της `cut`, ανεξάρτητα από την επιτυχία της `cat` (αν και στην περίπτωση του `cat /etc/passwd` πάντα επιτυγχάνει εκτός αν δεν υπάρχει το αρχείο).

Σημαντικό

- Να θυμάστε ότι οι εντολές `deluser` και `groupdel` είναι **μόνιμες**. Πριν από τη χρήση τους, βεβαιωθείτε ότι έχετε δημιουργήσει **αντίγραφα ασφαλείας** ή έχετε λάβει τα απαραίτητα μέτρα για να αποφύγετε απώλεια δεδομένων.
- Η χρήση των εντολών διαχείρισης χρηστών και ομάδων απαιτεί **δικαιώματα root**. Μπορείτε να χρησιμοποιήσετε την εντολή `sudo` για να εκτελέσετε τις εντολές με δικαιώματα root.

Διαχείριση Πακέτων

Ο παρακάτω πίνακας περιλαμβάνει βασικές και προχωρημένες εντολές για τη διαχείριση πακέτων σε συστήματα Linux Debian. Οι εντολές αυτές είναι κρίσιμες για τη σωστή εγκατάσταση, αναβάθμιση και αφαίρεση λογισμικού, καθώς και για τη διαχείριση των εξαρτήσεων που απαιτούνται για τη λειτουργία των πακέτων. Καλύπτουν μια ευρεία γκάμα λειτουργιών, από την απλή εγκατάσταση και αφαίρεση πακέτων μέχρι πιο σύνθετες διαδικασίες, όπως η επαναδιαμόρφωση και η διαχείριση πηγών πακέτων.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
--------	-----------	-------------------------	---------

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
apt update	Ενημερώνει τη λίστα των διαθέσιμων πακέτων.	-	Κατεβάζει τις τελευταίες πληροφορίες για τα διαθέσιμα πακέτα από τις ρυθμισμένες πηγές.
apt upgrade	Αναβαθμίζει όλα τα εγκατεστημένα πακέτα.	-	Εγκαθιστά τις τελευταίες εκδόσεις των πακέτων που είναι ήδη εγκατεστημένα.
apt install <πακέτο>	Εγκαθιστά ένα νέο πακέτο.	<πακέτο> : Το όνομα του πακέτου που θέλετε να εγκαταστήσετε.	Εγκαθιστά το συγκεκριμένο πακέτο και τις εξαρτήσεις του.
apt remove <πακέτο>	Αφαιρεί ένα εγκατεστημένο πακέτο.	<πακέτο> : Το όνομα του πακέτου που θέλετε να αφαιρέσετε.	Αφαιρεί το συγκεκριμένο πακέτο, αλλά διατηρεί τις ρυθμίσεις του και τα αρχεία διαμόρφωσης.
apt purge <πακέτο>	Αφαιρεί ένα πακέτο και τις ρυθμίσεις του.	<πακέτο> : Το όνομα του πακέτου που θέλετε να αφαιρέσετε.	Αφαιρεί το πακέτο και όλες τις ρυθμίσεις του, καθαρίζοντας πλήρως το σύστημα.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
apt search <όρος>	Αναζητά πακέτα που ταιριάζουν με τον όρο.	<όρος> : Ο όρος αναζήτησης για τα πακέτα.	Επιστρέφει μια λίστα πακέτων που περιέχουν τον καθορισμένο όρο στην περιγραφή ή το όνομα.
apt show <πακέτο>	Εμφανίζει λεπτομέρειες για ένα πακέτο.	<πακέτο> : Το όνομα του πακέτου για το οποίο θέτετε λεπτομέρειες.	Παρέχει πληροφορίες όπως η έκδοση, οι εξαρτήσεις και η περιγραφή του πακέτου.
apt autoremove	Αφαιρεί πακέτα που δεν χρειάζονται πια.	-	Αφαιρεί πακέτα που εγκαταστάθηκαν ως εξαρτήσεις και δεν χρησιμοποιούνται πλέον από κανένα εγκατεστημένο πακέτο.
apt list --installed	Εμφανίζει όλα τα εγκατεστημένα πακέτα.	-	Παρέχει μια λίστα με όλα τα πακέτα που είναι εγκατεστημένα στο σύστημα.
apt-cache policy <πακέτο>	Εμφανίζει πληροφορίες σχετικά με τις εκδόσεις ενός πακέτου.	<πακέτο> : Το όνομα του πακέτου για το οποίο θέτετε πληροφορίες.	Δείχνει τις διαθέσιμες εκδόσεις του πακέτου και από ποιες πηγές προέρχονται.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
<code>apt full-upgrade</code>	Αναβαθμίζει το σύστημα, διαχειριζόμενο τις εξαρτήσεις.	-	Αναβαθμίζει όλα τα πακέτα, επιτρέποντας την αφαίρεση πακέτων αν είναι απαραίτητο για την επίλυση εξαρτήσεων και την αναβάθμιση.
<code>apt edit-sources</code>	Επεξεργάζεται το αρχείο πηγών πακέτων.	-	Ανοίγει το αρχείο <code>/etc/apt/sources.list</code> (ή άλλα στο <code>/etc/apt/sources.list.d/</code>) για να προσθέσετε ή να τροποποιήσετε πηγές.
<code>dpkg -i <πακέτο.deb></code>	Εγκαθιστά ένα πακέτο από αρχείο <code>.deb</code> .	<code><πακέτο.deb></code> : Το αρχείο του πακέτου που θέτετε να εγκαταστήσετε.	Εγκαθιστά το πακέτο απευθείας από το αρχείο <code>.deb</code> , χωρίς να χρησιμοποιεί το APT, και δεν επιλύει αυτόματα εξαρτήσεις.
<code>dpkg -r <πακέτο></code>	Αφαιρεί ένα πακέτο χωρίς να διαγράψει τις ρυθμίσεις του.	<code><πακέτο></code> : Το όνομα του πακέτου που θέτετε να αφαιρέσετε.	Αφαιρεί το πακέτο, διατηρώντας τα αρχεία ρυθμίσεων του για μελλοντική επαναφορά.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
<code>dpkg --get-selections</code>	Εμφανίζει όλα τα εγκατεστημένα πακέτα.	-	Παρέχει μια λίστα με όλα τα εγκατεστημένα πακέτα στο σύστημα και την κατάστασή τους (π.χ. install, deinstall).
<code>dpkg-reconfigure <πακέτο></code>	Επαναδιαμορφώνει ένα εγκατεστημένο πακέτο.	<πακέτο> : Το όνομα του πακέτου που θέτετε να επαναδιαμορφώσετε.	Ξανατρέχει τη διαδικασία ρύθμισης του πακέτου, επιτρέποντας την αλλαγή ρυθμίσεων που έγιναν κατά την εγκατάσταση.

▼ Παραδείγματα Εντολών

- Εγκατάσταση του tilix:

```
1 | sudo apt install tilix | sh
```

- Αφαίρεση του tilix:

```
1 | sudo apt remove tilix | sh
```

- Αναζήτηση για πακέτα που περιέχουν "tilix":

```
1 | apt search tilix | sh
```

- Λεπτομέρειες για το tilix:

```
1 | apt show tilix | sh
```

- Εγκατάσταση του tilix από .deb αρχείο:

```
1 | sudo dpkg -i tilix_1.9.0_amd64.deb | sh
```

- Αφαίρεση του tilix χωρίς διαγραφή ρυθμίσεων:

1

```
sudo dpkg -r tilix
```

sh

- Εμφάνιση όλων των εγκατεστημένων πακέτων:

1

```
dpkg --get-selections
```

sh

- Επαναδιαμόρφωση του tilix:

1

```
sudo dpkg-reconfigure tilix
```

sh

Βασικό

- Χρησιμοποιήστε την εντολή `sudo` πριν από κάθε εντολή `apt` ή `dpkg` που τροποποιεί το σύστημα, για να έχετε πρόσβαση ως `root` και να μπορέσετε να διαχειριστείτε τα πακέτα.
- Τα `.deb` αρχεία μπορούν να κατεβάσουν από την επίσημη ιστοσελίδα του εκάστοτε λογισμικού ή από άλλες έμπιστες πηγές.
- Πριν από την εγκατάσταση ή την αφαίρεση οποιουδήποτε πακέτου, συνιστάται να δημιουργήσετε **αντίγραφα ασφαλείας**, για να έχετε μια ρεζέρβα σε περίπτωση προβλημάτων.

Διαχείριση Πόρων

Αυτός ο πίνακας παρέχει μια συνοπτική επισκόπηση των πιο συνηθισμένων εντολών για τη διαχείριση πόρων σε συστήματα Linux. Οι εντολές αυτές μπορούν να χρησιμοποιηθούν για να παρακολουθείτε τη χρήση CPU, μνήμης, δίσκου και δικτύου, και να αναγνωρίσετε οποιαδήποτε θέματα απόδοσης ή προβλήματα πόρων.

Βασικές Εντολές

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
--------	-----------	-------------------------	---------

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
top	Εμφανίζει σε πραγματικό χρόνο τη χρήση πόρων του συστήματος και τις διεργασίες.	-u <user> (φιλτράρισμα κατά χρήστη)	Παρέχει δυναμικές πληροφορίες για τη χρήση CPU, μνήμης, swar, τον αριθμό των διεργασιών, τον uptime του συστήματος και τη μέση φόρτο. Είναι ένα ισχυρό εργαλείο για άμεση παρακολούθηση.
htop	Εναλλακτική του top με γραφικό και διαδραστικό περιβάλλον.	-u <user> (φιλτράρισμα κατά χρήστη)	Μια βελτιωμένη έκδοση του top που προσφέρει ευκολότερη διαχείριση διεργασιών με χρήση χρωμάτων και γραφικών γραμμών προόδου. Απαιτείται εγκατάσταση (<code>sudo apt install htop</code>).
free	Εμφανίζει τη χρήση μνήμης του συστήματος.	-h (ανθρώπινα αναγνώσιμη μορφή)	Δείχνει τη συνολική, χρησιμοποιούμενη και διαθέσιμη φυσική μνήμη (RAM) και μνήμη swar, σε ανθρώπινα αναγνώσιμη μορφή (π.χ., GB, MB).
df	Εμφανίζει τη χρήση δίσκου για τα συστήματα αρχείων.	-h (ανθρώπινα αναγνώσιμη μορφή)	Παρέχει πληροφορίες για τη διαθέσιμη και χρησιμοποιούμενη χωρητικότητα των προσαρτημένων (mounted) συστημάτων αρχείων, συμπεριλαμβανομένου του ποσοστού χρήσης και του σημείου προσάρτησης.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
du	Εμφανίζει τη χρήση δίσκου για συγκεκριμένα αρχεία/ φακέλους.	-h (ανθρώπινα αναγνώσιμη μορφή), -s (συνοπτικά)	Χρήσιμο για την ανάλυση της χωρητικότητας που καταλαμβάνουν αρχεία και φάκελοι σε ένα συγκεκριμένο μονοπάτι. Το -s δίνει το συνολικό μέγεθος του φακέλου.
ps	Εμφανίζει τις τρέχουσες διεργασίες.	aux (όλες οι διεργασίες, χρήστης, τερματικό), ef (πλήρες φορμάτ)	Παρέχει στατικές πληροφορίες (στιγμιότυπο) για τις διεργασίες που τρέχουν στο σύστημα. Οι επιλογές aux ή ef είναι κοινές για να δουν όλες τις διεργασίες με λεπτομέρειες.

Προχωρημένες Εντολές

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
vmstat	Εμφανίζει πληροφορίες για τη μνήμη, CPU, διεργασίες και I/O σε πραγματικό χρόνο.	-s (συνοπτική αναφορά), delay count (επαναληπτική έξοδο)	Παρέχει στατιστικά για τη συνολική απόδοση του συστήματος, συμπεριλαμβανομένων των διεργασιών, της μνήμης, των I/O block, των διακοπών και της δραστηριότητας της CPU. Είναι χρήσιμο για την ανίχνευση bottlenecks.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
iostat	Εμφανίζει στατιστικά για τη χρήση δίσκου και CPU.	-x (εκτενής αναφορά), interval count	Παρακολουθεί την απόδοση των συσκευών I/O (σκληρών δίσκων) και τη χρήση της CPU. Το -x παρέχει εκτενείς στατιστικές, όπως ουρά I/O και χρόνους εξυπηρέτησης, ενώ το interval δίνει συνεχή ενημέρωση.
netstat	Εμφανίζει πληροφορίες για τις συνδέσεις δικτύου, πίνακες δρομολόγησης και στατιστικά διεπαφών.	-tuln (TCP/ UDP, listening, αριθμητικές διευθύνσεις)	Χρήσιμο για την παρακολούθηση των ενεργών δικτυακών συνδέσεων (εισερχόμενες/εξερχόμενες), των θυρών που ακούνε για συνδέσεις, και την ανάλυση προβλημάτων δικτύου. Η επιλογή n εμφανίζει αριθμητικές διευθύνσεις αντί για ονόματα host.
sar	Συλλέγει, αναφέρει και αποθηκεύει στατιστικά για τη χρήση πόρων του συστήματος.	-u (CPU), -r (μνήμη), -d (δίσκος)	Μέρος του πακέτου sysstat, το sar είναι ένα ισχυρό εργαλείο για την παρακολούθηση της απόδοσης σε βάθος χρόνου. Συλλέγει δεδομένα σε τακτά χρονικά διαστήματα και μπορεί να τα αναφέρει ή να τα αποθηκεύσει για μελλοντική ανάλυση, βοηθώντας στον εντοπισμό τάσεων και προβλημάτων.

▼ Παραδείγματα Εντολών

Βασικές Εντολές

- Εμφανίζει τις διεργασίες που εκτελεί ο χρήστης "juniormin".

```
1 top -u juniormin sh
```

- Εμφανίζει σε γραφικό περιβάλλον τις διεργασίες του χρήστη "juniormin".

```
1 htop -u juniormin sh
```

- Εμφανίζει τη χρήση μνήμης σε ανθρώπινα αναγνώσιμη μορφή.

```
1 free -h sh
```

- Εμφανίζει τη χρήση δίσκου σε ανθρώπινα αναγνώσιμη μορφή.

```
1 df -h sh
```

- Εμφανίζει τη χρήση δίσκου για τον φάκελο "/home/juniormin" σε ανθρώπινα αναγνώσιμη μορφή.

```
1 du -sh /home/juniormin sh
```

Σημείωση: Προσθήκη `-s` για συνοπτική εμφάνιση του μεγέθους του φακέλου.

- Εμφανίζει όλες τις διεργασίες με λεπτομέρειες.

```
1 ps aux sh
```

Προχωρημένες Εντολές

- Εμφανίζει συνοπτικά στατιστικά για μνήμη, CPU και I/O.

```
1 vmstat -s sh
```

- Εμφανίζει εκτενείς πληροφορίες για τη χρήση δίσκου και CPU.

```
1 iostat -x sh
```

- Εμφανίζει πληροφορίες για όλες τις συνδέσεις TCP/UDP που ακούνε.

```
1 netstat -tuln sh
```

- Συλλέγει και εμφανίζει στατιστικά για την χρήση CPU ανά δευτερόλεπτο (για 5 φορές).

```
1 sar -u 1 5 sh
```

- Συλλέγει και εμφανίζει στατιστικά για την χρήση μνήμης ανά δευτερόλεπτο (για 5

φορές).

1

```
sar -r 1 5
```

sh

Σημείωση:

- Αντικαταστήστε το "juniormin" με το όνομα χρήστη που σας ενδιαφέρει.
- Οι εντολές htop , iostat , netstat , sar ενδέχεται να χρειάζονται εγκατάσταση μέσω του διαχειριστή πακέτων (sudo apt install <όνομα_πακέτου> , π.χ., sudo apt install htop sysstat net-tools).

Διαχείριση Αρχείων και Φακέλων

Βασικές και προχωρημένες εντολές που είναι απαραίτητες για τη διαχείριση αρχείων και φακέλων σε συστήματα Linux Debian. Οι εντολές αυτές καλύπτουν ένα ευρύ φάσμα λειτουργιών, από την απλή πλοήγηση και την αντιγραφή αρχείων μέχρι την αναζήτηση και την τροποποίηση δικαιωμάτων πρόσβασης. Η καλή γνώση αυτών των εντολών είναι κρίσιμη για κάθε διαχειριστή συστημάτων, καθώς διευκολύνει την αποτελεσματική διαχείριση και συντήρηση του συστήματος, επιτρέποντας την εκτέλεση καθημερινών εργασιών με ευχέρεια και ασφάλεια.

Βασικές Εντολές

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
ls	Εμφανίζει τα περιεχόμενα ενός φακέλου.	-l (λεπτομερής), -a (κρυφά), -h (αναγνώσιμα μεγέθη)	Εμφανίζει αρχεία και υποφακέλους. Το -l δίνει πληροφορίες για δικαιώματα, ιδιοκτήτη, μέγεθος, ημερομηνία. Το -a εμφανίζει κρυφά αρχεία (. prefix). Το -h εμφανίζει μεγέθη σε KB, MB, GB.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
cd	Αλλάζει τον τρέχοντα φάκελο εργασίας.	<φάκελος> , .. , ~	Μεταβαίνει στον καθορισμένο φάκελο. .. μεταβαίνει στον γονικό φάκελο. ~ μεταβαίνει στον προσωπικό φάκελο του χρήστη.
pwd	Εμφανίζει το τρέχοντα φάκελο εργασίας.	-	Εμφανίζει το πλήρες μονοπάτι του φακέλου στον οποίο βρίσκεται ο χρήστης.
cp	Αντιγράφει αρχεία ή φακέλους.	-r (φακέλους αναδρομικά), -i (επιβεβαίωση), -u (μόνο νεότερα), -a (αρχεία/φακέλους διατηρώντας ιδιότητες)	Αντιγράφει αρχεία και φακέλους. Το -a είναι ιδανικό για backups καθώς διατηρεί δικαιώματα, χρονικές σημάνσεις, κ.λπ.
mv	Μετακινεί ή μετονομάζει αρχεία ή φακέλους.	-i (επιβεβαίωση), -u (μόνο νεότερα)	Μετακινεί αρχεία από ένα σημείο σε άλλο ή αλλάζει το όνομά τους στον ίδιο φάκελο.
rm	Διαγράφει αρχεία ή φακέλους.	-r (αναδρομικά), -f (αναγκαστική), -i (επιβεβαίωση)	Διαγράφει αρχεία ή φακέλους. Χρειάζεται προσοχή, ειδικά με το -f που παρακάμπτει προειδοποιήσεις.
mkdir	Δημιουργεί νέο φάκελο.	-p (γονικούς φακέλους αν δεν υπάρχουν)	Δημιουργεί ένα ή περισσότερους φακέλους.
rmdir	Διαγράφει κενό φάκελο.	-	Διαγράφει μόνο αν ο φάκελος είναι κενός.

Εντολή	Περιγραφή	Ορίσματα/ Παράμετροι	Ανάλυση
touch	Δημιουργεί κενό αρχείο ή αλλάζει χρονικές σημάνσεις.	<όνομα_αρχείου>	Δημιουργεί ένα νέο κενό αρχείο ή ενημερώνει την ημερομηνία τροποποίησης ενός υπάρχοντος αρχείου.
cat	Εμφανίζει το περιεχόμενο ενός αρχείου.	<όνομα_αρχείου>	Διαβάζει και εμφανίζει το περιεχόμενο ενός αρχείου απευθείας στο τερματικό.
less	Εμφανίζει το περιεχόμενο ενός αρχείου με δυνατότητα κύλισης.	<όνομα_αρχείου>	Επιτρέπει την προβολή μεγάλων αρχείων χωρίς να φορτώνει ολόκληρο το αρχείο στη μνήμη. Παρέχει δυνατότητα αναζήτησης και κύλισης.

Προχωρημένες Εντολές

Εντολή	Περιγραφή	Ορίσματα/Παράμετροι	Ανάλυση
find	Αναζητά αρχεία και φάκελους με βάση κριτήρια.	<κατηγορία> , -name , -type f/d , -size , - mtime	Ισχυρή εντολή για την εύρεση αρχείων με βάση το όνομα, τον τύπο (f για αρχείο, d για φάκελο), το μέγεθος, την ημερομηνία τροποποίησης, κ.λπ. Μπορεί να συνδυαστεί με ενέργειες (-exec).

Εντολή	Περιγραφή	Ορίσματα/Παράμετροι	Ανάλυση
grep	Αναζητά κείμενο μέσα σε αρχεία.	-i (αγνοεί περίπτωση), -r (αναδρομικά), -v (γραμμές που δεν ταιριάζουν), -l (μόνο ονόματα αρχείων)	Χρήσιμη για την εύρεση συγκεκριμένων συμβολοσειρών κειμένου μέσα σε ένα ή περισσότερα αρχεία. Ιδανική για αναζήτηση logs ή αρχείων διαμόρφωσης.
tar	Συμπιέζει ή αποσυμπιέζει αρχεία/ φακέλους.	-cvf (δημιουργία, εμφάνιση, αρχείο), -xvf (εξαγωγή, εμφάνιση, αρχείο), -z (gzip), -j (bzip2), -J (xz)	Δημιουργεί αρχεία αρχειοθέτησης (.tar) ή τα αποσυμπιέζει. Οι επιλογές -z , -j , -J προσθέτουν συμπίεση.
chmod	Αλλάζει τα δικαιώματα πρόσβασης σε αρχεία/ φακέλους.	<δικαιώματα> (π.χ., 755 , u+rwx), <αρχείο/ φάκελος>	Καθορίζει ποιος (χρήστης, ομάδα, άλλοι) μπορεί να διαβάσει (r), να γράψει (w) ή να εκτελέσει (x) ένα αρχείο. Μπορεί να χρησιμοποιηθεί με αριθμητική (octal) ή συμβολική μορφή.
chown	Αλλάζει τον ιδιοκτήτη και την ομάδα ενός αρχείου/ φακέλου.	<ιδιοκτήτης>:<ομάδα> <αρχείο/φάκελος> , -R (αναδρομικά)	Καθορίζει ποιος είναι ο ιδιοκτήτης και ποια ομάδα ανήκει το αρχείο ή ο φάκελος. Το -R αλλάζει αναδρομικά την ιδιοκτησία σε υποφακέλους και αρχεία.

Εντολή	Περιγραφή	Ορίσματα/Παράμετροι	Ανάλυση
ln	Δημιουργεί συνδέσμους (σκληρούς/συμβολικούς) σε αρχεία/φακέλους.	-s (συμβολικός σύνδεσμος)	Δημιουργεί συνδέσμους προς άλλα αρχεία ή φακέλους. Οι συμβολικοί σύνδεσμοι (-s , σαν συντομεύσεις) είναι οι πιο συνηθισμένοι.

▼ Παραδείγματα Εντολών

Βασικές Εντολές:

- Εμφάνιση περιεχομένου φακέλου (με λεπτομέρειες, κρυφά αρχεία, αναγνώσιμα μεγέθη):

```
1 ls -lah sh
```

- Μετάβαση στον φάκελο Documents :

```
1 cd Documents sh
```

- Μετάβαση στον προσωπικό φάκελο:

```
1 cd ~ sh
```

- Εμφάνιση τρέχοντος φακέλου:

```
1 pwd sh
```

- Αντιγραφή του αρχείου text.txt στον φάκελο Downloads :

```
1 cp text.txt Downloads/ sh
```

- Αντιγραφή του φακέλου my_project στον φάκελο backups διατηρώντας ιδιότητες:

```
1 cp -a my_project backups/ sh
```

- Μετονομασία του αρχείου text.txt σε new_text.txt :

```
1 mv text.txt new_text.txt sh
```

- Διαγραφή του αρχείου `new_text.txt` :

```
1 rm new_text.txt sh
```

- Δημιουργία νέου φακέλου με όνομα `new_folder` και υποφάκελο `sub_folder` :

```
1 mkdir -p new_folder/sub_folder sh
```

- Διαγραφή του κενού φακέλου `empty_folder` :

```
1 rmdir empty_folder sh
```

- Δημιουργία νέου κενού αρχείου `my_file.txt` :

```
1 touch my_file.txt sh
```

- Εμφάνιση περιεχομένου του αρχείου `README.md` :

```
1 cat README.md sh
```

- Προβολή μεγάλου αρχείου `log` με δυνατότητα κύλισης:

```
1 less /var/log/syslog sh
```

Προχωρημένες Εντολές:

- Αναζήτηση για όλα τα αρχεία με επέκταση `.txt` στον τρέχοντα φάκελο και τους υποφακέλους του:

```
1 find . -name "*.txt" sh
```

- Αναζήτηση για φακέλους με όνομα `"my_folder"`:

```
1 find . -type d -name "my_folder" sh
```

- Αναζήτηση για κείμενο `"error"` σε όλα τα αρχεία του τρέχοντος φακέλου αναδρομικά:

```
1 grep -r "error" . sh
```

- Συμπίεση όλων των αρχείων στον τρέχοντα φάκελο σε ένα αρχείο `archive.tar.gz` :

```
1 tar -czvf archive.tar.gz . sh
```

- Αποσυμπίεση του αρχείου `archive.tar.gz` :

```
1 tar -xzvf archive.tar.gz sh
```

- Αλλαγή δικαιωμάτων στο `script.sh` ώστε ο ιδιοκτήτης να έχει πλήρη δικαιώματα (read, write, execute), και η ομάδα/άλλοι μόνο ανάγνωση και εκτέλεση (r-x):

```
1 chmod 755 script.sh sh
```

- Αλλαγή ιδιοκτητή του `file.txt` σε `juniormin` και ομάδας σε `developers` :

```
1 sudo chown juniormin:developers file.txt sh
```

- Δημιουργία συμβολικού συνδέσμου `link_to_file.txt` προς το αρχείο `file.txt` :

```
1 ln -s file.txt link_to_file.txt sh
```

Σημείωση:

- Η τελεία `.` αναφέρεται στον τρέχοντα φάκελο.
- Ο αστερίσκος `*` αναφέρεται σε όλα τα αρχεία και τους φακέλους στον τρέχοντα φάκελο (με προσοχή, καθώς μπορεί να συμπεριλάβει και κρυφά αρχεία).
- Πολλές εντολές διαχείρισης αρχείων, ειδικά αυτές που αφορούν την αλλαγή ιδιοκτησίας ή δικαιωμάτων σε συστήματα αρχείων, απαιτούν **δικαιώματα root** (χρήση `sudo`).

Για να ελέγξετε τη λειτουργία κάθε εντολής, μπορείτε να χρησιμοποιήσετε το όρισμα `-h` ή `--help` για να εμφανίσετε βοήθεια.

Εξωτερικές πηγές

[Debian Manpages](#)

Linux Permissions

👤 Ευαγγελία Κατσιαντώνη 📅 10/18/24 🌐 Linux 🔑 permissions, administration ⌚ Περίπου 19 λεπτά

Εισαγωγή

Από αρχάριους έως έμπειρους επαγγελματίες, αυτός ο περιεκτικός οδηγός αποτελεί μια απαραίτητη πηγή για την εμβάθυνση ή την ανανέωση των γνώσεών σας στα δικαιώματα αρχείων στο Linux, προσφέροντας σαφήνεια και πρακτικές εφαρμογές.

Τι είναι τα Permissions

Φανταστείτε το Linux ως ένα μεγάλο κτίριο, το οποίο στεγάζει τα **κεντρικά γραφεία μιας εταιρείας**. Σε αυτά τα γραφεία, κάθε **χώρος** (π.χ., ένα προσωπικό γραφείο, μια αίθουσα συσκέψεων, ο χώρος υποδοχής κ.ο.κ) αντιπροσωπεύει έναν **φάκελο** (directory) στο Linux. Μέσα σε κάθε χώρο, υπάρχουν διάφορα **αντικείμενα εξοπλισμού** και **πληροφοριακό υλικό** – όπως υπολογιστές, τηλέφωνα, καρέκλες, **ατομικοί φάκελοι προσωπικού, παραστατικά, έγγραφα κανονισμών λειτουργίας** και άλλα στοιχεία – που αντιπροσωπεύουν τα **αρχεία** (files) στο Linux.

Σε ένα τέτοιο εταιρικό περιβάλλον, η ομαλή λειτουργία και η ασφάλεια βασίζονται σε έναν σαφή **κανονισμό λειτουργίας**. Αυτός ο κανονισμός καθορίζει ποιος έχει το δικαίωμα να μπει σε κάθε χώρο και τι μπορεί να κάνει με τον εξοπλισμό ή τα πληροφοριακά στοιχεία που βρίσκονται εκεί. Με βάση αυτόν τον κανονισμό, τίθεται σε εφαρμογή ένα **σύστημα ελέγχου πρόσβασης**, το οποίο διασφαλίζει ότι οι κανόνες τηρούνται αυστηρά.

Θυμάμαι!

Αυτό το σύστημα, που διέπει τις αλληλεπιδράσεις των ανθρώπων με τους χώρους και τον εξοπλισμό τους, αντιστοιχεί ακριβώς στα **permissions** (δικαιώματα) του Linux, τα οποία ρυθμίζουν την αλληλεπίδραση των χρηστών με τα αρχεία και τους φακέλους.

Κάθε φορά που κάποιος προσπαθεί να αποκτήσει πρόσβαση σε ένα χώρο ή ένα αντικείμενο, το σύστημα ελέγχου πρόσβασης εξετάζει δύο βασικά ερωτήματα:

- Ποιος είναι αυτός που ζητάει πρόσβαση;

- Είναι ένας **συγκεκριμένος υπάλληλος** (ο κάτοχος / **owner**), ο οποίος έχει τον κύριο λόγο στον προσωπικό του χώρο;
 - Είναι μέλος μιας **ομάδας εργασίας** (του **group**), που μοιράζεται κοινούς χώρους και εργαλεία;
 - Ή είναι οποιοσδήποτε **άλλος επισκέπτης** (**others**) της εταιρείας, με περιορισμένες αρμοδιότητες;
- Τι ακριβώς επιτρέπεται να κάνει μέσα;
 - Για ένα γραφείο (φάκελος για το Linux):
 - Να κοιτάξετε ποια αντικείμενα περιέχει (δικαίωμα ανάγνωσης - r).
 - Να τακτοποιήσετε ή να προσθέσετε νέα αντικείμενα, ή να αφαιρέσετε υπάρχοντα (δικαίωμα εγγραφής - w).
 - Να μπειτε μέσα του και να τον χρησιμοποιήσετε ως βάση για τις εργασίες σας (δικαίωμα εκτέλεσης - x).
 - Για ένα αντικείμενο (αρχείο για το Linux):
 - Να διαβάσετε ένα έγγραφο (δικαίωμα ανάγνωσης - r).
 - Να τροποποιήσετε ή να διαγράψετε ένα έγγραφο (δικαίωμα εγγραφής - w).
 - Να χρησιμοποιήσετε έναν προτζέκτορα για μια παρουσίαση (δικαίωμα εκτέλεσης - x, όπου η "εκτέλεση" είναι η λειτουργία του αντικειμένου).

Αναλογία Δικαιωμάτων

Ας δούμε πώς μεταφράζονται αυτά τα δικαιώματα σε συγκεκριμένα παραδείγματα του εταιρικού κτιρίου, σε χώρους και εξοπλισμό:

- Το Γραφείο του Διευθυντή (Linux: Αντιστοιχεί στον Προσωπικό φάκελο /home/διευθυντής)
 - Αυτό είναι το **προσωπικό γραφείο** του διευθυντή. Μόνο αυτός έχει πλήρη πρόσβαση και έλεγχο/χρήση στα πάντα.
 - Οι υπάλληλοι (ομάδα) μπορούν να **δουν την επιγραφή της πόρτας** και την **ύπαρξη του γραφείου**, αλλά **δεν μπορούν να μπουν μέσα** χωρίς πρόσκληση ή άδεια.
 - Οι επισκέπτες (άλλοι) δεν έχουν καμία πρόσβαση.

Μετάφραση σε Linux permissions

- **Δικαιώματα:** drwxr----- (ή d740)
 - d : Είναι φάκελος.
 - rwx : Ο **Διευθυντής** έχει πλήρη δικαιώματα (ανάγνωση, εγγραφή, εκτέλεση).
 - r-- : Η **Ομάδα** έχει δικαιώματα ανάγνωσης.
 - --- : Οι **Άλλοι** δεν έχουν κανένα δικαίωμα.

- **Το Γραφείο του Υπαλλήλου** (Linux: Αντιστοιχεί στον Προσωπικό φάκελο /home/υπάλληλος)

- Αντίστοιχα, αυτό είναι το προσωπικό γραφείο ενός υπαλλήλου. Ο υπάλληλος έχει πλήρη έλεγχο-χρήση στο χώρο και στα αντικείμενα που βρίσκονται σε αυτόν.
- Οι συνάδελφοι (ομάδα) μπορούν να **δουν τι υπάρχει μέσα** στο γραφείο (π.χ. να κοιτάζουν από το παράθυρο). Εάν είναι ο υπάλληλος μέσα, μπορούν να **εισέλθουν**, να **καθίσουν στην καρέκλα επισκεπτών** και να διαβάσουν τα ενημερωτικά φυλλάδια που υπάρχουν στο τραπέζακι, **χωρίς να οικειοποιηθούν ή να καταστρέψουν κάτι**.
- Οι επισκέπτες (άλλοι) δεν έχουν καμία πρόσβαση στον προσωπικό του χώρο.

Μετάφραση σε Linux permissions

- **drwxr-x---** (ή d750)
 - d : Είναι φάκελος.
 - rwx : Ο **Υπάλληλος** έχει πλήρη δικαιώματα (ανάγνωση, εγγραφή, εκτέλεση).
 - r-x : Η **Ομάδα** του υπαλλήλου έχει δικαιώματα ανάγνωσης (οπτική παρατήρηση από το παράθυρο) και εκτέλεσης (εισόδου, καθίσματος στην καρέκλα, ανάγνωσης εντύπων), αλλά όχι εγγραφής (να σημειώσουν στα έντυπα ή να καταστρέψουν).
 - --- : Οι **Άλλοι** δεν έχουν κανένα δικαίωμα.

- **Η Αίθουσα Συσκέψεων** (Linux: Αντιστοιχεί στον Κοινόχρηστο φάκελο /shared/meetings)

- Αυτή είναι μια αίθουσα όπου συγκεκριμένες ομάδες εργασίας συναντιούνται και συνεργάζονται.
- Οι **προτζέκτορες** και τα **έγγραφα** (αρχεία) που βρίσκονται εκεί είναι για χρήση από την ομάδα. Ένα μέλος της ομάδας μπορεί να **δείξει** κάτι στον προτζέκτορα (εκτέλεση) ή να τροποποιήσει έγγραφα, αλλά ένας τυχαίος επισκέπτης όχι.

Μετάφραση σε Linux permissions

- **drwxrwsr-x** (ή **d2775**)
 - **d** : Είναι φάκελος.
 - **rw**x : Ο **Ιδιοκτήτης** (π.χ. ο διαχειριστής της αίθουσας) έχει πλήρη δικαιώματα (ανάγνωση, εγγραφή, εκτέλεση).
 - **rws** : Η **Ομάδα** (π.χ. "Ομάδα Πωλήσεων") έχει δικαιώματα ανάγνωσης, εγγραφής, εκτέλεσης, και το **s** (SetGID) σημαίνει ότι κάθε νέο αρχείο/φάκελος που δημιουργείται εδώ θα ανήκει αυτόματα στην ομάδα "Ομάδα Πωλήσεων".
 - **r-x** : Οι **Άλλοι** (π.χ. επισκέπτες) μπορούν να δουν την αίθουσα και ίσως να μπουν (εάν είναι ανοιχτή), αλλά δεν μπορούν να γράψουν ή να τροποποιήσουν κάτι.

- **Ο Χώρος Υποδοχής** (*Linux: Αντιστοιχεί στον Ορατό σε όλους φάκελο, π.χ. /var/www/html για webserver*)
 - Αυτό είναι το σημείο όπου όλοι μπορούν να έχουν πρόσβαση, να δουν πληροφορίες (π.χ. περιηγητές/browsers), αλλά δεν μπορούν να αλλάξουν τίποτα.

Μετάφραση σε Linux permissions

- **drwxr-xr-x** (ή **d755**)
 - **d** : Είναι φάκελος.
 - **rw**x : Ο **Κάτοχος** (π.χ. ο υπεύθυνος της υποδοχής) έχει πλήρη δικαιώματα (ανάγνωση, εγγραφή, εκτέλεση).
 - **r-x** : Η **Ομάδα** και οι **Άλλοι** μπορούν να διαβάσουν (read) και να μπουν (execute) στον χώρο, αλλά όχι να γράψουν ή να τροποποιήσουν κάτι.

- **Ο Κεντρικός Διακομιστής Κλιματισμού** (*Linux: Αντιστοιχεί σε αρχείο ρύθμισης συστήματος, π.χ. /etc/systemd/system/ac_unit.conf*)
 - Αυτός ο διακομιστής είναι σαν ένα κρίσιμο αρχείο ρύθμισης που επηρεάζει όλο το κτίριο.
 - Μόνο ο κεντρικός διαχειριστής του κτιρίου μπορεί να το διαβάσει και να το τροποποιήσει, ενώ μια ειδική ομάδα διαχειριστών (π.χ. της συντήρησης) μπορεί μόνο να το διαβάσει. Οι υπόλοιποι δεν έχουν καμία πρόσβαση.

Μετάφραση σε Linux permissions

- `-rw-r-----` (ή `640`)
 - `-` : Είναι αρχείο.
 - `rw-` : Ο **Κάτοχος** (π.χ. ο `root` / ο Γενικός Διευθυντής) μπορεί να το διαβάσει και να το τροποποιήσει.
 - `r--` : Μια ειδική **Ομάδα Διαχειριστών** (π.χ. "HVAC_Admins") μπορεί να το διαβάσει.
 - `---` : Οι **Άλλοι** δεν έχουν καμία πρόσβαση.

Ο Χρήστης Root

Ο Αποκλειστικός Κύριος του Κτιρίου

Σε αυτή την αναλογία, ο χρήστης `root` στο Linux είναι ο **απόλυτος ιδιοκτήτης** και **διαχειριστής του κτιρίου** (της εταιρείας). Είναι σαν τον **Πρόεδρο της εταιρείας**, ο οποίος έχει την εξουσία να:

- Ανοίξει οποιαδήποτε πόρτα.
- Αλλάξει οποιοδήποτε κλειδί.
- Μετακινήσει ή να αλλάξει την ιδιοκτησία οποιουδήποτε αντικειμένου.
- Δώσει ή να αφαιρέσει εξουσίες από οποιονδήποτε υπάλληλο ή ομάδα.

Όταν ένας διευθυντής χρειάζεται να κάνει μια κρίσιμη αλλαγή που ξεπερνά τα δικά του δικαιώματα, όπως να εγκαταστήσει ένα νέο σύστημα ασφαλείας ή να αλλάξει τη δομή των γραφείων, πρέπει να "ζητήσει" την άδεια του Προέδρου. Στο Linux, αυτό αντιστοιχεί στη χρήση της εντολής `sudo` (**SuperUser Do**). Όταν χρησιμοποιούμε `sudo`, ζητάμε ουσιαστικά από το σύστημα να εκτελέσει την εντολή με τα δικαιώματα του `root`, δηλαδή με την απόλυτη εξουσία του Προέδρου της εταιρείας.

Γιατί Είναι Ζωτικής Σημασίας;

Η σημασία των δικαιωμάτων (permissions) στο Linux εκτείνεται πέρα από την απλή οργάνωση αρχείων. Αποτελούν το **θεμέλιο** πάνω στο οποίο οικοδομείται η **ασφάλεια** και η **σταθερότητα** ολόκληρου του λειτουργικού συστήματος. Η κρισιμότητα των permissions προκύπτει από τους ακόλουθους λόγους:

- **Ασφάλεια:** Τα σωστά permissions αποτρέπουν τη **μη εξουσιοδοτημένη πρόσβαση** και τη χειραγώγηση ευαίσθητων δεδομένων ή κρίσιμων αρχείων συστήματος. Χωρίς την

ορθή εφαρμογή τους, ένας κακόβουλος χρήστης θα μπορούσε να αποκτήσει πρόσβαση, να τροποποιήσει ή ακόμα και να διαγράψει σημαντικά αρχεία, θέτοντας σε κίνδυνο την **ακεραιότητα** και την **εμπιστευτικότητα** των πληροφοριών σας.

- **Οργανωτική Συνοχή:** Μέσω των permissions, διασφαλίζεται ότι κάθε χρήστης ή εφαρμογή έχει πρόσβαση **μόνο στους πόρους που είναι απαραίτητοι** για την εκτέλεση της εκάστοτε εργασίας. Αυτό ελαχιστοποιεί το ρίσκο λαθών, συγκρούσεων και ανεπιθύμητων ενεργειών, διατηρώντας το σύστημα σε **εύρυθμη** και **οργανωμένη** λειτουργία.
- **Λειτουργική Αποδοτικότητα:** Πολλά προγράμματα και υπηρεσίες απαιτούν **συγκεκριμένα δικαιώματα** για την ορθή τους λειτουργία. Λανθασμένες ρυθμίσεις permissions μπορούν να οδηγήσουν σε σφάλματα, αδυναμία εκτέλεσης εφαρμογών ή ακόμα και σε δυσλειτουργίες ολόκληρου του συστήματος. Η **σωστή διαχείριση** των δικαιωμάτων εξασφαλίζει την **απρόσκοπτη εκτέλεση** όλων των διαδικασιών.

Κατηγορίες Χρηστών

Ποιος Έχει Πρόσβαση;

Στο Linux, τα permissions ορίζονται για τρεις συγκεκριμένες κατηγορίες "προσώπων":

- **Κάτοχος (Owner - u):** Αυτός είναι ο χρήστης που δημιούργησε το αρχείο ή τον φάκελο, ή στον οποίο έχει εκχωρηθεί η ιδιοκτησία του. Έχει συνήθως τον πλήρη έλεγχο.
- **Ομάδα (Group - g):** Μια ομάδα χρηστών στους οποίους ανήκει το αρχείο ή ο φάκελος. Όλα τα μέλη αυτής της ομάδας μοιράζονται τα ίδια δικαιώματα πρόσβασης στο συγκεκριμένο αντικείμενο. Αυτό είναι φανταστικό για συνεργασία σε projects!
- **Άλλοι (Others - o):** Όλοι οι υπόλοιποι χρήστες του συστήματος που δεν είναι ο κάτοχος και δεν ανήκουν στην ομάδα του αρχείου. Συνήθως έχουν τα πιο περιορισμένα δικαιώματα.

Τύποι Δικαιωμάτων

Τι Μπορεί να Κάνει;

Για κάθε μία από τις παραπάνω κατηγορίες, μπορούμε να ορίσουμε τρία διαφορετικά δικαιώματα:

- **Ανάγνωση (Read - r):**

- **Για αρχεία:** Σας επιτρέπει να διαβάσετε το περιεχόμενό τους.
- **Για φακέλους:** Σας επιτρέπει να **προβάλετε τα περιεχόμενα** του φακέλου, δηλαδή να δείτε ποια αρχεία και υποφακέλους περιέχει.

- **Εγγραφή (Write - w):**

- **Για αρχεία:** Μπορείτε να τροποποιήσετε ή να διαγράψετε το αρχείο.
- **Για φακέλους:** Μπορείτε να δημιουργήσετε, να διαγράψετε, να μετονομάσετε ή να μετακινήσετε αρχεία και υποφακέλους **μέσα** σε αυτόν τον φάκελο.

Προσοχή:

Χωρίς δικαίωμα εκτέλεσης (x) στον φάκελο, το δικαίωμα εγγραφής (w) δεν έχει νόημα.

- **Εκτέλεση (Execute - x):**

- **Για αρχεία:** Σας επιτρέπει να εκτελέσετε το αρχείο, αν πρόκειται για εκτελέσιμο πρόγραμμα ή script.
- **Για φακέλους:** Σας επιτρέπει να **πλοηγηθείτε** μέσα στον φάκελο, δηλαδή να χρησιμοποιήσετε την εντολή `cd` για να τον εισέλθετε και να αποκτήσετε πρόσβαση στα αρχεία και τους υποφακέλους του.

Σημαντικό:

Να σημειωθεί ότι αυτό το δικαίωμα είναι απαραίτητο για την προσπέλαση του περιεχομένου ενός καταλόγου, ακόμα κι αν δεν υπάρχει δικαίωμα ανάγνωσης (r) για την εμφάνιση των περιεχομένων με την εντολή `ls` .

Αριθμητική Αναπαράσταση Δικαιωμάτων

Για την απλοποιημένη και αποδοτική διαχείριση των δικαιωμάτων πρόσβασης στο Linux, χρησιμοποιείται ένα **οκταδικό αριθμητικό σύστημα**. Σε αυτό το σύστημα, κάθε μεμονωμένο δικαίωμα αντιστοιχεί σε μια συγκεκριμένη αριθμητική τιμή:

- **Ανάγνωση (r):** 4
- **Εγγραφή (w):** 2
- **Εκτέλεση (x):** 1
- **Κανένα (-):** 0

Για να προσδιορίσετε την συνολική αριθμητική τιμή για μια συγκεκριμένη κατηγορία χρηστών (Κάτοχος, Ομάδα, Άλλοι), αρκεί να **αθροίσετε** τις τιμές των δικαιωμάτων που επιθυμείτε να της αποδώσετε. Το τελικό αποτέλεσμα θα είναι ένας μοναδικός αριθμός που αντιπροσωπεύει το σύνολο των δικαιωμάτων για αυτή την κατηγορία.

Σημαντικό!

Ένας συνδυασμός δικαιωμάτων δίνεται με **τρεις αριθμούς**:

- ο **πρώτος** αφορά τον **Κάτοχο** (Owner - u),
- ο **δεύτερος** την **Ομάδα** (Group - g),
- και ο **τρίτος** τους **Άλλους** (Others - o)

Αντιστοιχία Αναπαράστασης Συμβολικών και Αριθμητικών Δικαιωμάτων

Συμβολική	Περιγραφή	Αριθμητική
rwX	Ανάγνωση, Εγγραφή, Εκτέλεση	7
rw-	Ανάγνωση, Εγγραφή	6
r-x	Ανάγνωση, Εκτέλεση	5
r--	Ανάγνωση	4
-wX	Εγγραφή, Εκτέλεση	3
-w-	Εγγραφή	2
--x	Εκτέλεση	1
---	Κανένα	0

Παραδείγματα

- 777 : Όλοι έχουν πλήρη δικαιώματα (rwX).

Προσοχή! Σπάνια χρησιμοποιείται και θεωρείται επικίνδυνο.

- 644 : Ο κάτοχος έχει r/w, ενώ η ομάδα και οι άλλοι μόνο r.

Συνήθης ρύθμιση για αρχεία κειμένου.

- 755 : Ο κάτοχος έχει r/w/x, ενώ η ομάδα και οι άλλοι έχουν r/x.

Συνήθης ρύθμιση για εκτελέσιμα αρχεία και φακέλους.

- 751 : Ο κάτοχος έχει r/w/x, η ομάδα έχει r/x και οι άλλοι έχουν r/x.

Γραφική επίλυση προβλήματος

Έστω ότι θέλουμε να υπολογίσουμε τα δικαιώματα που πρέπει να εκχωρήσουμε στο αρχείο file έτσι ώστε ο κάτοχος να έχει πλήρη έλεγχο, η ομάδα να έχει δικαίωμα ανάγνωσης και εκτέλεσης και τέλος όλοι οι υπόλοιποι μόνο δικαίωμα εκτέλεσης.



Έστω ότι θέλουμε να υπολογίσουμε τα δικαιώματα που πρέπει να εκχωρήσουμε στο αρχείο **file** έτσι ώστε ο **κάτοχος** να έχει **πλήρη έλεγχο**, η **ομάδα** να έχει δικαίωμα **ανάγνωσης και εκτέλεσης** και τέλος **όλοι οι υπόλοιποι** μόνο δικαίωμα **εκτέλεσης**.

u g o

Βήμα-1°

Σε ένα πρόχειρο δημιουργώ κενές τις 'τριάδες' για τους τύπους αδειών (rwX) για τις ομάδες (u,g,o)

Βήμα-2°

r w X r X X

Συμπληρώνω για κάθε ομάδα τις άδειες (δικαιώματα)

Βήμα-3°

4 2 1 4 1 1

Αποκωδικοποιώ (αντικαθιστώ) τις άδειες (δικαιώματα) των ομάδων με τις αντίστοιχες δυαδικές

Βήμα-4°

7 5 1

Αθροίζω τις άδειες (δικαιώματα) της κάθε ομάδας

Συντάσσω την εντολή **chmod** και την πληκτρολογώ στο τερματικό

chmod 751 file

```
admin@pc-test:~$ chmod 751 file
```

Έλεγχος Δικαιωμάτων

Μπορείτε να δείτε τα permissions ενός αρχείου/φακέλου, χρησιμοποιώντας την εντολή `ls -l`.

1

```
ls -l my_script.sh
```

sh

η έξοδος στο τερματικό σας, θα είναι κάτι παρόμοιο με το παρακάτω:

```
user@hostname:~# ls -l my_script.sh
-rwxr-xr-- 1 user group 1024 May 22 10:00 my_script.sh
user@hostname:~#
```

Ανάλυση του `-rwxr-xr--` :

Θέση	Χαρακτήρας/Ομάδα	Περιγραφή
1	- (ή <code>d</code> , <code>l</code>)	Τύπος: - (αρχείο), <code>d</code> (φάκελος), <code>l</code> (symlink).
2-4	<code>rwX</code>	Κάτοχος δικαιώματα.
5-7	<code>r-X</code>	Ομάδα δικαιώματα.
8-10	<code>r--</code>	Άλλοι δικαιώματα.

Αλλαγή Δικαιωμάτων

Για να αλλάξετε δικαιώματα (permissions) σε αρχείο / φάκελο εκτελέστε την εντολή `chmod` .

1. Αριθμητική Μορφή

Ο Πιο Συνηθισμένος Τρόπος

Σύνταξη: `chmod [αριθμοί] [αρχείο/φάκελος]`

Παραδείγματα:

```
user@hostname:~# chmod 754 my_file.txt
user@hostname:~# ls -l my_file.txt
-rwxr-xr-- 1 user group 0 May 25 15:00 my_file.txt
user@hostname:~# chmod 755 my_script.sh
user@hostname:~# ls -l my_script.sh
-rwxr-xr-x 1 user group 0 May 25 15:01 my_script.sh
user@hostname:~# chmod 600 config.conf
user@hostname:~# ls -l config.conf
```

```
-rw----- 1 user group 0 May 25 15:02 config.conf  
user@hostname:~#
```

2. Συμβολική Μορφή

Για Συγκεκριμένες Αλλαγές

- `u` , `g` , `o` , `a` (όλοι)
- `+` (πρόσθεσε), `-` (αφαίρεσε), `=` (όρισε)
- `r` , `w` , `x`

Σύνταξη: `chmod [ποιός][+/-/=][τι] [αρχείο/φάκελος]`

Παραδείγματα:

```
user@hostname:~# chmod u+x my_script.sh  
user@hostname:~# ls -l my_script.sh  
-rwxr-xr-x 1 user group 0 May 25 15:03 my_script.sh  
user@hostname:~# chmod o-w important_doc.txt  
user@hostname:~# ls -l important_doc.txt  
-rw-r--r-- 1 user group 0 May 25 15:04 important_doc.txt  
user@hostname:~# chmod go=r public_info.txt  
user@hostname:~# ls -l public_info.txt  
-rw-r--r-- 1 user group 0 May 25 15:05 public_info.txt  
user@hostname:~# chmod -R a-x .  
user@hostname:~#
```

Αλλαγή Ιδιοκτησίας

chown & *chgrp*

Κάθε αρχείο/φάκελος έχει ιδιοκτήτη χρήστη και ομάδα.

chown : Αλλαγή Ιδιοκτήτη

Αλλάζει τον ιδιοκτήτη χρήστη και/ή την ιδιοκτήτρια ομάδα.

Σύνταξη: `chown [νέος_χρήστης]:[νέα_ομάδα] [αρχείο/φάκελος]`

Παραδείγματα:

```
root@CT-001-DEVSRV-UBU-2404:~# chown john report.pdf
```

```
root@CT-001-DEVSRV-UBU-2404:~# ls -l report.pdf
-rw-r--r-- 1 john group 0 May 25 15:06 report.pdf
root@CT-001-DEVSRV-UBU-2404:~# chown admin:developers data_folder
root@CT-001-DEVSRV-UBU-2404:~# ls -ld data_folder
drwxr-xr-x 2 admin developers 4096 May 25 15:07 data_folder
root@CT-001-DEVSRV-UBU-2404:~# chown -R webmaster:www-data /var/www/html
root@CT-001-DEVSRV-UBU-2404:~#
```

Σημαντικό!

Η `chown` απαιτεί **root δικαιώματα** (`sudo`). Χρησιμοποιήστε την με προσοχή.

'chgrp': Αλλαγή Ομάδας (λιγότερο συχνή)

Αλλάζει μόνο την ιδιοκτήτρια ομάδα.

Σύνταξη: `chgrp [νέα_ομάδα] [αρχείο/φάκελος]`

Παράδειγμα:

```
root@CT-001-DEVSRV-UBU-2404:~# chgrp team_alpha shared_doc.odt
root@CT-001-DEVSRV-UBU-2404:~# ls -l shared_doc.odt
-rw-r--r-- 1 user team_alpha 0 May 25 15:08 shared_doc.odt
root@CT-001-DEVSRV-UBU-2404:~#
```

Ειδικά Δικαιώματα

Επέκταση των Ελέγχων Πρόσβασης UNIX/Linux

Πέρα από τα βασικά δικαιώματα ανάγνωσης (`r`), εγγραφής (`w`), και εκτέλεσης (`x`), το σύστημα αρχείων του Linux (και γενικότερα των συστημάτων τύπου UNIX) παρέχει τρία ειδικά δικαιώματα. Αυτά τα δικαιώματα προσφέρουν **επιπρόσθετα επίπεδα ελέγχου** και λειτουργικότητας, επιτρέποντας πιο εξελιγμένες ρυθμίσεις ασφαλείας και διαχείρισης. Η κατανόηση και η σωστή εφαρμογή τους είναι ζωτικής σημασίας για τη διασφάλιση της ακεραιότητας και της ασφάλειας του συστήματος, καθώς η λανθασμένη χρήση τους μπορεί να δημιουργήσει σημαντικά κενά ασφαλείας.

1. SetUID (Set User ID)

- **Αριθμητική Τιμή:** `4000` (προστίθεται ως το πρώτο ψηφίο στην οκταδική αναπαράσταση).

- **Συμβολική Αναπαράσταση:** Εμφανίζεται ως `s` στη θέση του δικαιώματος εκτέλεσης (`x`) για τον **κάτοχο** (`u`). Αν το αρχείο δεν έχει δικαίωμα εκτέλεσης για τον κάτοχο, εμφανίζεται ως κεφαλαίο `S` (`rwsr-xr-x` αν έχει `x` ή `rwsr-xr-x` αν δεν έχει `x`).
- **Λειτουργία:** Όταν ένα εκτελέσιμο αρχείο έχει ορισμένο το SetUID bit, εκτελείται με τα δικαιώματα του **ιδιοκτήτη του αρχείου** (owner), ανεξάρτητα από τον χρήστη που το εκκινεί. Το πιο κλασικό παράδειγμα είναι η εντολή `passwd`, η οποία επιτρέπει σε έναν απλό χρήστη να αλλάξει τον κωδικό πρόσβασής του, παρόλο που η τροποποίηση του `/etc/shadow` (όπου αποθηκεύονται οι κωδικοί) απαιτεί δικαιώματα `root`. Η εντολή `passwd` ανήκει στον `root` και έχει SetUID, επιτρέποντας την προσωρινή αναβάθμιση προνομίων κατά την εκτέλεσή της.
- **Σημασία για την Ασφάλεια:** Το SetUID bit είναι ένα ισχυρό χαρακτηριστικό, αλλά απαιτεί **μεγάλη προσοχή**. Η λανθασμένη εφαρμογή του σε μη ασφαλή προγράμματα μπορεί να αποτελέσει σοβαρό κενό ασφαλείας, επιτρέποντας σε έναν κακόβουλο χρήστη να αποκτήσει αυξημένα προνόμια.

2. SetGID (Set Group ID)

- **Αριθμητική Τιμή:** `2000` (προστίθεται ως το πρώτο ψηφίο στην οκταδική αναπαράσταση).
- **Συμβολική Αναπαράσταση:** Εμφανίζεται ως `s` στη θέση του δικαιώματος εκτέλεσης (`x`) για την **ομάδα** (`g`). Αν το αρχείο δεν έχει δικαίωμα εκτέλεσης για την ομάδα, εμφανίζεται ως κεφαλαίο `S` (`rwxr-sr-x` αν έχει `x` ή `rwxr-Sr-x` αν δεν έχει `x`).
- **Λειτουργία:**
 - **Σε Αρχεία:** Όταν ένα εκτελέσιμο αρχείο έχει ορισμένο το SetGID bit, εκτελείται με τα δικαιώματα της **ομάδας στην οποία ανήκει το αρχείο**, ανεξάρτητα από την κύρια ομάδα του χρήστη που το εκτελεί.
 - **Σε Φακέλους:** Αυτή είναι η πιο συχνή και πρακτική χρήση του SetGID. Όταν ορίζεται σε έναν φάκελο, οποιαδήποτε **νέα αρχεία ή υποφακέλοι** που δημιουργούνται μέσα σε αυτόν τον φάκελο θα **κληρονομήσουν αυτόματα την ομάδα του γονικού φακέλου** (parent directory), αντί της κύριας ομάδας του χρήστη που τα δημιούργησε. Αυτό είναι εξαιρετικά χρήσιμο για την διευκόλυνση της συνεργασίας σε κοινόχρηστους καταλόγους, διασφαλίζοντας ότι όλα τα μέλη μιας ομάδας έχουν αυτόματα πρόσβαση στα νέα αρχεία.

3. Sticky Bit

- **Αριθμητική Τιμή:** `1000` (προστίθεται ως το πρώτο ψηφίο στην οκταδική αναπαράσταση).
- **Συμβολική Αναπαράσταση:** Εμφανίζεται ως `t` στη θέση του δικαιώματος

εκτέλεσης (*x*) για τους **άλλους** (*o*). Αν ο φάκελος δεν έχει δικαίωμα εκτέλεσης για τους άλλους, εμφανίζεται ως κεφαλαίο *T* (*rwXrwxrwt* αν έχει *x* ή *rwXrwxT* αν δεν έχει *x*).

- **Λειτουργία:** Το Sticky Bit εφαρμόζεται **αποκλειστικά σε φακέλους**. Όταν ένας φάκελος έχει ορισμένο το Sticky Bit, οι χρήστες που έχουν δικαιώματα εγγραφής (*w*) σε αυτόν τον φάκελο μπορούν να δημιουργήσουν νέα αρχεία και υποφακέλους. Ωστόσο, μπορούν να **διαγράψουν ή να μετονομάσουν μόνο τα αρχεία ή τους υποφακέλους που έχουν δημιουργήσει οι ίδιοι** ή αυτά που ανήκουν σε αυτούς. Δεν μπορούν να διαγράψουν ή να μετονομάσουν αρχεία άλλων χρηστών, ακόμα κι αν έχουν δικαιώματα εγγραφής στον φάκελο.
- **Κλασικό Παράδειγμα:** Ο φάκελος `/tmp` . Αυτός ο φάκελος είναι παγκοσμίως εγγράψιμος (*rwXrwxrwt*), επιτρέποντας σε όλους τους χρήστες να δημιουργούν προσωρινά αρχεία. Ωστόσο, χάρη στο Sticky Bit, ένας χρήστης δεν μπορεί να διαγράψει τα προσωρινά αρχεία ενός άλλου χρήστη, ενισχύοντας την ασφάλεια και την ακεραιότητα του συστήματος.

Ορισμός Ειδικών Δικαιωμάτων με `chmod`

Ο ορισμός των ειδικών δικαιωμάτων γίνεται με την εντολή `chmod` , προσθέτοντας την αντίστοιχη αριθμητική τιμή (*4* , *2* , ή *1*) ως το **τέταρτο, αριστερότερο ψηφίο** στην οκταδική αναπαράσταση, πριν από τους τρεις βασικούς αριθμούς (ιδιοκτήτης, ομάδα, άλλοι).

Σύνταξη: `chmod [ειδικό_δικαίωμα][αριθμοί_δικαιωμάτων] [αρχείο/φάκελος]`

Παραδείγματα:

```
root@CT-001-DEVSRV-UBU-2404:~# chmod 4755 my_program
root@CT-001-DEVSRV-UBU-2404:~# ls -l my_program
-rwsr-xr-x 1 root root 1024 May 25 15:09 my_program
root@CT-001-DEVSRV-UBU-2404:~# chmod 2770 shared_folder
root@CT-001-DEVSRV-UBU-2404:~# ls -ld shared_folder
drwxrws--- 2 user group 4096 May 25 15:10 shared_folder
root@CT-001-DEVSRV-UBU-2404:~# chmod 1777 public_uploads
root@CT-001-DEVSRV-UBU-2404:~# ls -ld public_uploads
drwxrwxrwt 2 user group 4096 May 25 15:11 public_uploads
root@CT-001-DEVSRV-UBU-2404:~#
```

Συνοπτικός Πίνακας Δικαιωμάτων

Βασικές Τιμές

Αριθμητική	Συμβολική	Περιγραφή Δικαιωμάτων
7	<code>rx</code>	Πλήρης Έλεγχος
6	<code>rw</code>	Ανάγνωση & Εγγραφή
5	<code>r-x</code>	Ανάγνωση & Εκτέλεση
4	<code>r--</code>	Μόνο Ανάγνωση
0	<code>---</code>	Κανένα Δικαίωμα
777	<code>rxrxrx</code>	Όλοι πλήρη. Αποφύγετέ το!
755	<code>rxr-xr-x</code>	Κάτοχος: Πλήρης Έλεγχος, Ομάδα/Άλλοι: Ανάγνωση & Εκτέλεση.
644	<code>rw-r--r--</code>	Κάτοχος: Ανάγνωση & Εγγραφή, Ομάδα/Άλλοι: Μόνο Ανάγνωση.
600	<code>rw-----</code>	Κάτοχος: Ανάγνωση & Εγγραφή, Ομάδα/Άλλοι: Κανένα Δικαίωμα.
4755	<code>rwsr-xr-x</code>	SetUID: Εκτελεί ως ιδιοκτήτης.
2770	<code>rxrws---</code>	SetGID: Κληρονομιά ομάδας για φακέλους.
1777	<code>rxrxrwt</code>	Sticky Bit: Μόνο ο κάτοχος διαγράφει.

Βέλτιστες Πρακτικές & Συχνά Λάθη

- **Αρχή Ελάχιστου Προνόμιου:** Δώστε τα ελάχιστα απαραίτητα δικαιώματα. Ποτέ 777 χωρίς λόγο!
- **Φάκελοι vs. Αρχεία:** Να θυμάστε τις διαφορές των `r`, `w`, `x` σε αρχεία και φακέλους.
- **Ειδικά Δικαιώματα:** Χρησιμοποιήστε τα με **απόλυτη προσοχή** και μόνο όταν είναι απαραίτητο.
- **Αναδρομικές Αλλαγές (`-R`):** Χρησιμοποιήστε τις προσεκτικά σε κρίσιμες διαδρομές. Πάντα ελέγξτε δύο φορές!

Εν Κατακλείδι

Πλέον, κατέχετε τις θεμελιώδεις γνώσεις για να κατανοήσετε και να διαχειριστείτε τα δικαιώματα στο Linux. Έχετε εξοικειωθεί με τις κατηγορίες χρηστών, τους βασικούς τύπους δικαιωμάτων, την αριθμητική τους αναπαράσταση και τις κρίσιμες εντολές `chmod`, `chown` και `chgrp`. Αυτά τα εργαλεία είναι απαραίτητα για να λειτουργήσετε ως αποτελεσματικοί και ασφαλείς διαχειριστές συστήματος.

Βιβλιογραφία

1. *The Linux Foundation - Classic SysAdmin: Understanding Linux File Permissions:* <https://www.linuxfoundation.org/blog/blog/classic-sysadmin-understanding-linux-file-permissions>
2. *Understanding Linux File Permissions:* <https://www.redhat.com/sysadmin/linux-file-permissions-explained>
3. *DevAnswers - How to View File and Folder Permissions in Ubuntu Linux Command Line:* <https://devanswe.rs/how-to-view-file-and-folder-permissions-in-ubuntu/>
4. *Pluralsight - How to change directory permissions in Linux with chmod:* <https://www.pluralsight.com/resources/blog/tech-operations/linux-file-permissions>
5. *Go Linux Cloud - 15 most used chown command in Linux:* <https://www.golinuxcloud.com/chown-command-in-linux/>
6. *Go Linux Cloud - 5+ practical examples to learn sticky bit in Linux:* <https://www.golinuxcloud.com/sticky-bit-linux/>
7. *ls(1) - Linux man page:* <https://man7.org/linux/man-pages/man1/ls.1.html>
8. *chmod(1) - Linux man page:* <https://man7.org/linux/man-pages/man1/chmod.1.html>
9. *chown(1) - Linux man page:* <https://man7.org/linux/man-pages/man1/chown.1.html>
0. *chown(2) - LinuxTect chown Command Tutorial:* <https://linuxtect.com/chown-command-tutorial/>
1. *chgrp(1) - Linux man page:* <https://man7.org/linux/man-pages/man1/chgrp.1.html>

Επιλεγμένα - Τοπικά αρχεία

-  [Classic SysAdmin- Understanding Linux File Permissions](#)
-  [FileSystem Permissions No1](#)
-  [FileSystem Permissions No2](#)

-  [Permissions:Command chown No1](#)
-  [Permissions:Command chown No2](#)
-  [Permissions:Learn-sticky-bit](#)

Shell scripting

Ευαγγελία Κατσιαντώνη 10/28/24 Linux code, scripting Περίπου 11 λεπτά

Εισαγωγή

Το **Shell scripting** είναι ένας βασικός μηχανισμός αυτοματοποίησης στον τομέα της πληροφορικής, ιδιαίτερα σε περιβάλλοντα που βασίζονται στο λειτουργικό σύστημα UNIX/Linux. Αποτελεί ένα εργαλείο που επιτρέπει την εκτέλεση εντολών σε αλληλουχία για την υλοποίηση σύνθετων διαδικασιών με χρήση ενός κέλυφους (shell), όπως το **Bash**, το **Zsh**, ή το **Ksh**.

Τι είναι το Shell Scripting;

Το Shell scripting αναφέρεται στη συγγραφή σεναρίων (scripts) που εκτελούνται από έναν διερμηνέα κελύφους. Ένα script είναι ένα αρχείο κειμένου που περιλαμβάνει διαδοχικές εντολές γραμμής εντολών, οι οποίες μπορούν να συνδυαστούν με δομές προγραμματισμού για τη δημιουργία αυτοματοποιημένων διαδικασιών.

Το Shell scripting επιτρέπει την αλληλεπίδραση με το λειτουργικό σύστημα σε χαμηλό επίπεδο και χρησιμοποιείται ευρέως για αυτοματοποίηση, διαχείριση συστημάτων και ανάπτυξη λογισμικού.

Τεχνική Υλοποίηση

Με την έκφραση "Τεχνική Υλοποίηση", αναφερόμαστε στη σύνταξη ενός Shell script. Αυτή εξαρτάται από τον τύπο του κελύφους που χρησιμοποιείται (π.χ., Bash, Zsh, Sh). Ανεξάρτητα από το περιβάλλον, τα βασικά στοιχεία περιλαμβάνουν:

- Εντολές Κελύφους:** Τα shell scripts περιλαμβάνουν εντολές που εκτελούνται από το κέλυφος (shell) του λειτουργικού συστήματος. Αυτές οι εντολές μπορεί να περιλαμβάνουν εντολές για τη διαχείριση αρχείων, την εκτέλεση προγραμμάτων και την εκτέλεση συστημικών εργασιών.

```
1 # Δημιουργία ενός νέου φακέλου
2 mkdir my_folder
```

sh

- Σύνταξη:** Η σύνταξη των shell scripts διαφέρει ανάλογα με το κέλυφος που

χρησιμοποιείται (π.χ., Bash, Zsh, Sh). Κάθε κέλυφος έχει τις δικές του ιδιαιτερότητες και δυνατότητες.

```
1 # Ένα απλό script σε Bash sh
2 #!/bin/bash
3 echo "Hello, World!"
```

3. **Μεταβλητές:** Μπορείτε να δηλώσετε και να χρησιμοποιήσετε μεταβλητές για να αποθηκεύσετε δεδομένα και να τα χρησιμοποιήσετε σε διάφορες εντολές. Οι μεταβλητές μπορούν να περιέχουν κείμενο, αριθμούς ή αποτελέσματα εντολών.

```
1 # Δημιουργία και χρήση μεταβλητής sh
2 name="Alice"
3 echo "Hello, $name!"
```

4. **Δομές Ελέγχου:** Τα shell scripts υποστηρίζουν δομές ελέγχου όπως `if`, `for`, `while`, και `case`, που επιτρέπουν την εκτέλεση εντολών υπό συγκεκριμένες συνθήκες ή την επανάληψη εντολών.

```
1 # Χρήση δομής if sh
2 number=10
3 if [ $number -gt 5 ]; then
4     echo "$number είναι μεγαλύτερο από 5"
5 fi
```

5. **Σχόλια:** Μπορείτε να προσθέσετε σχόλια στα scripts σας χρησιμοποιώντας το σύμβολο `#`. Τα σχόλια είναι χρήσιμα για την τεκμηρίωση του κώδικα και την εξήγηση των λειτουργιών του.

```
1 # Αυτό είναι ένα σχόλιο sh
2 echo "Αυτό το script εκτυπώνει ένα μήνυμα"
```

6. **Είσοδος και Έξοδος:** Τα shell scripts μπορούν να διαχειρίζονται είσοδο και έξοδο μέσω της κονσόλας ή αρχείων. Μπορείτε να διαβάσετε δεδομένα από το χρήστη ή να γράψετε αποτελέσματα σε αρχεία.

```
1 # Ανάγνωση εισόδου από το χρήστη sh
2 read -p "Ποιο είναι το όνομά σας; " username
3 echo "Γεια σου, $username!"
```

7. **Σφάλματα και Διαχείριση Σφαλμάτων:** Είναι σημαντικό να διαχειρίζεστε τα

σφάλματα στα scripts σας, χρησιμοποιώντας εντολές όπως `exit` και `trap` για να ελέγχετε την εκτέλεση και να χειρίζεστε απρόβλεπτες καταστάσεις.

```
1  # Έλεγχος αν ένα αρχείο υπάρχει sh
2  filename="myfile.txt"
3  if [ -e $filename ]; then
4      echo "$filename υπάρχει."
5  else
6      echo "$filename δεν υπάρχει."
7      exit 1
8  fi
```

8. Επαναχρησιμοποίηση Κώδικα: Μπορείτε να δημιουργήσετε συναρτήσεις μέσα στα scripts σας για να επαναχρησιμοποιείτε κώδικα και να οργανώνετε καλύτερα τη λογική του script.

```
1  # Δημιουργία συνάρτησης sh
2  function greet {
3      echo "Γειά σου, $1!"
4  }
5
6  # Κλήση της συνάρτησης
7  greet "Bob"
```

Tips

Τα scripts ξεκινούν με τη γραμμή **shebang** `#!/bin/sh` (ή `#!/bin/bash` για Bash), που δηλώνει τον διερμηνέα του script.

Υποστηριζόμενα Περιβάλλοντα

Τα Shell scripts εκτελούνται σε όλα τα περιβάλλοντα που διαθέτουν έναν διερμηνέα κελύφους:

- **Linux:** Όλες οι μεγάλες διανομές (Debian, Red Hat, Ubuntu).
- **macOS:** Περιλαμβάνει υποστήριξη για Bash και Zsh.
- **Windows:** Μέσω του Windows Subsystem for Linux ([WSL](#)) ή εργαλείων όπως το Git Bash και το Cygwin.

Απαιτήσεις Γνώσεων

Για την αποτελεσματική συγγραφή Shell scripts, απαιτούνται:

- Καλή γνώση της γραμμής εντολών:** Εντολές όπως `ls`, `cd`, `cat`, `grep`, `echo`, `touch` κ.α.
- Κατανόηση εννοιών προγραμματισμού:** Μεταβλητές, συνθήκες, επαναλήψεις.
- Γνώση των εργαλείων UNIX/Linux:** Για την ενίσχυση της λειτουργικότητας των scripts.

Κύριες Εφαρμογές των Shell Scripts

Αυτοματοποίηση Εργασιών (Αντίγραφα Ασφαλείας και Διαχείριση Αρχείων)

Η αυτοματοποίηση επαναλαμβανόμενων εργασιών, όπως η δημιουργία αντιγράφων ασφαλείας και η διαχείριση αρχείων, αποτελεί έναν από τους πιο συνηθισμένους και σημαντικούς τομείς εφαρμογής του Shell scripting. Με τη χρήση εντολών όπως `tar`, `gzip` και `rsync`, τα scripts μπορούν να δημιουργούν αντίγραφα ασφαλείας δεδομένων αυτόματα, να τα συμπιέζουν και να τα μεταφέρουν σε απομακρυσμένους servers ή εξωτερικά μέσα αποθήκευσης. Επιπλέον, παρέχουν δυνατότητες για την προγραμματισμένη εκτέλεση, εξασφαλίζοντας την προστασία των δεδομένων χωρίς την ανάγκη ανθρώπινης παρέμβασης. Στη διαχείριση αρχείων, το Shell scripting αυτοματοποιεί διαδικασίες όπως η οργάνωση, η διαγραφή, και η μετονομασία μεγάλων αριθμών αρχείων, χρησιμοποιώντας εντολές όπως `mv`, `find` και `rm`. Αυτό επιτρέπει την αποδοτική διαχείριση πόρων σε συστήματα παραγωγής και ανάπτυξης. Συνολικά, το Shell scripting βελτιώνει την αποδοτικότητα, μειώνει τον κίνδυνο σφαλμάτων και αυξάνει την αξιοπιστία των κρίσιμων εργασιών συντήρησης σε σύγχρονα συστήματα.

Διαχείριση Συστημάτων (Παρακολούθηση Logs και Έλεγχος Συστημάτων)

Τα logs, που αποτελούν βασική πηγή πληροφορίας για τη λειτουργία και την απόδοση του συστήματος, μπορούν να αναλυθούν και να φιλτραριστούν με εργαλεία όπως το `grep`, το `awk` και το `tail`. Για παράδειγμα, ένα script μπορεί να παρακολουθεί σε

πραγματικό χρόνο τα logs ενός server (`tail -f /var/log/syslog`) και να εντοπίζει σφάλματα ή σημαντικά γεγονότα με συγκεκριμένα πρότυπα αναζήτησης (`grep "ERROR"`). Επιπλέον, το Shell scripting επιτρέπει τον αυτοματοποιημένο έλεγχο συστήματος, όπως η παρακολούθηση της χρήσης πόρων (CPU, RAM, δίσκος) με εντολές όπως `top` , `df` , `free` ή `iostat` . Μπορούν επίσης να ρυθμιστούν ειδοποιήσεις (alerts) όταν κάποιος πόρος υπερβαίνει ένα προκαθορισμένο όριο. Τα σενάρια (scripts) αυτά είναι ιδανικά για τακτικούς ελέγχους, παραγωγή αναφορών ή ακόμα και για την αυτοματοποίηση επεμβάσεων (π.χ., επανεκκίνηση υπηρεσιών). Έτσι, το Shell scripting παρέχει μια ευέλικτη και προσαρμόσιμη πλατφόρμα για τη βελτιστοποίηση και την ασφάλεια συστημάτων.

Επεξεργασία Δεδομένων (Μετασχηματισμός Αρχείων Κειμένου με Χρήση Εργαλείων όπως το `sed`)

Η επεξεργασία δεδομένων μέσω Shell scripting είναι ιδιαίτερα αποδοτική για την αυτοματοποίηση μετασχηματισμών αρχείων κειμένου, χρησιμοποιώντας εργαλεία όπως το `sed` (stream editor). Το `sed` επιτρέπει την επεξεργασία αρχείων γραμμή-προς-γραμμή χωρίς να απαιτείται να τα ανοίξουμε σε περιβάλλον επεξεργασίας κειμένου. Μέσω εντολών όπως αντικατάσταση, διαγραφή ή εισαγωγή γραμμών, καθίσταται δυνατό να επεξεργαστούμε μεγάλους όγκους δεδομένων γρήγορα και με ακρίβεια. Για παράδειγμα, ένα script μπορεί να χρησιμοποιεί το `sed` για να αντικαταστήσει όλες τις εμφανίσεις μιας λέξης σε ένα αρχείο (`sed 's/παλιά_λέξη/νέα_λέξη/g' αρχείο.txt`), να αφαιρέσει περιττές γραμμές ή να μορφοποιήσει αρχεία καταγραφής (logs) για ανάλυση. Η δύναμη του `sed` έγκειται στην ικανότητά του να συνδυάζεται με άλλα εργαλεία όπως το `awk` , το `grep` και το `cut` , επιτρέποντας τον σύνθετο χειρισμό δεδομένων από διαφορετικές πηγές. Έτσι, το Shell scripting προσφέρει έναν ευέλικτο και επεκτάσιμο τρόπο διαχείρισης κειμένου για καθαρισμό, ανάλυση και προετοιμασία δεδομένων σε συστήματα και εφαρμογές.

Περιβάλλοντα Ανάπτυξης (Υλοποίηση DevOps Pipelines και Διαχείριση Containers)

Το Shell scripting διαδραματίζει καθοριστικό ρόλο στα σύγχρονα περιβάλλοντα ανάπτυξης λογισμικού, ειδικά στο πλαίσιο του **DevOps** και της διαχείρισης containerization τεχνολογιών όπως το **Docker**. Στο DevOps, τα scripts χρησιμοποιούνται για την αυτοματοποίηση των σταδίων ανάπτυξης, ενσωμάτωσης και παράδοσης λογισμικού (CI/CD pipelines). Αυτές οι διαδικασίες περιλαμβάνουν τη διαχείριση κώδικα, τη μεταγλώττιση, τη δοκιμή, τη δημιουργία artifacts και την τελική ανάπτυξη σε servers ή containers. Για παράδειγμα, ένα script μπορεί να τραβάει τον κώδικα από ένα

αποθετήριο Git, να το μεταγλωττίζει, να τρέχει αυτοματοποιημένα τεστ και να ανεβάζει τις επιτυχείς εκδόσεις σε μια πλατφόρμα όπως το Kubernetes. Παράλληλα, στον τομέα των containers, τα Shell scripts διευκολύνουν την αυτοματοποίηση στη δημιουργία, τη διαμόρφωση και την εκτέλεση container μέσω εργαλείων όπως το docker-compose ή εντολών Docker CLI. Αυτό περιλαμβάνει ενέργειες όπως η δημιουργία Docker images, η ανάπτυξη πολλαπλών containers με συνδεδεμένα δίκτυα, και η παρακολούθηση της κατάστασης των containers. Με αυτό τον τρόπο, το Shell scripting προσφέρει έναν εύκολο, επαναχρησιμοποιήσιμο και ευέλικτο τρόπο διαχείρισης κρίσιμων διεργασιών ανάπτυξης και υποδομών.

Βασικές Αρχές Συγγραφής Shell Scripts

1. Δομή και Σχολιασμός:

Τα scripts πρέπει να είναι ευανάγνωστα και πλήρως σχολιασμένα.

```
1 # Δημιουργία backup
2 tar -czf backup.tar.gz /path/to/data sh
```

2. Ασφάλεια:

Χρήση βέλτιστων πρακτικών, όπως η αποφυγή εντολών με δικαιώματα root χωρίς επαρκή έλεγχο.

3. Δοκιμή και Έλεγχος Σφαλμάτων:

Η χρήση εντολών όπως `set -e` και `set -u` συμβάλλει στον εντοπισμό σφαλμάτων κατά την εκτέλεση.

4. Απλότητα και Λειτουργικότητα:

Ο κώδικας πρέπει να είναι κατανοητός και εύκολα συντηρήσιμος.

Παράδειγμα: Δημιουργίας Αντιγράφου Ασφαλείας

Σενάριο

- Σε ένα σύστημα Ubuntu, Θέλουμε να αυτοματοποιήσουμε τη δημιουργία αντιγράφων ασφαλείας για το γονικό φάκελο του χρήστη **labrat**, σε μια δικτυακή θέση αποθήκευσης (NAS).
- Η ενέργεια αυτή θέλουμε να εκτελείται εβδομαδιαία κάθε Κυριακή στις 23:59 το βράδυ, ακόμα και αν ο υπολογιστής είναι απενεργοποιημένος.
- Το αρχείο πρέπει να είναι εύκολα αναγνωρίσιμο, σε περίπτωση αναζήτησης.

Η σχεδίαση υλοποίησης θα περιλαμβάνει το εργαλείο `cron job`. Για να καλύψουμε όμως την περίπτωση όπου ο υπολογιστής είναι απενεργοποιημένος τη δεδομένη στιγμή που πρέπει να εκτελεστεί το `script`, επιβάλλεται να χρησιμοποιήσουμε το `anacron` αντί για το `cron`. Το **`anacron`** είναι ιδανικό για τέτοιες περιπτώσεις, καθώς εκτελεί εργασίες που έχουν προγραμματιστεί με βάση το χρονικό διάστημα (π.χ., εβδομαδιαία, μηνιαία), ακόμα και αν ο υπολογιστής δεν ήταν ανοιχτός όταν έπρεπε να εκτελεστεί το `task`. Αυτό σημαίνει ότι το αντίγραφο ασφαλείας θα εκτελείται αμέσως μόλις ο υπολογιστής είναι σε λειτουργία, ακόμα και αν το χρονοδιάγραμμα του `cron` χάθηκε λόγω της απενεργοποίησης του συστήματος.

1. Δημιουργία του script

Αρχικά, πρέπει να συντάξουμε το script μας έτσι ώστε να μπορεί να εκτελείται μέσω `anacron`. Επειδή το `anacron` δεν χρησιμοποιεί τις ίδιες ρυθμίσεις όπως ακριβώς το `cron`, δεν χρειάζεται να προσδιορίσουμε μια συγκεκριμένη ώρα, αλλά μόνο την περιοδικότητα της εκτέλεσης. Σε κάθε τμήμα του κώδικα πρέπει να εισάγουμε αναλυτικές και ξεκάθαρες σημειώσεις, έτσι ώστε αυτά να είναι ευανάγνωστο, σε περίπτωση επιδιώρθωσης σφαλμάτων.

- **1.1** Με την επόμενη εντολή δημιουργούμε το αρχείο `/bin/labrat_backup_script.sh`

```
1 sudo nano /bin/labrat_backup_script.sh sh
```

- **1.2** Στο τερματικό εισάγουμε το επόμενο τμήμα κώδικα και αποθηκεύουμε τις αλλαγές:

```
1  #!/bin/bash sh
2
3  # Ορίζουμε το path του γονικού καταλόγου προς αντιγραφή, του χρήστη
4  labrat.
5  SOURCE_DIR="/home/labrat"
6
7  # Ορίζουμε το path του NAS storage (έστω π.χ είναι ο κατάλογος
8  nas_backups)
9  DEST_DIR="/mnt/nas_backups"
10
11 # Ορίζουμε το όνομα του αντιγράφου με ενσωμάτωση της ημερομηνίας και ώρας
12 δημιουργίας, έτσι ώστε αυτό να είναι αναγνωρίσιμο
13 BACKUP_NAME="backup_$(date +%d%m%Y%H%M%S).tar.gz"
14
15 # Δημιουργούμε τον κώδικα για το αντίγραφο ασφαλείας με την εντολή tar
16 tar -czf "$DEST_DIR/$BACKUP_NAME" "$SOURCE_DIR"
```

```

17
18 # Κάνουμε έλεγχο αν η δημιουργία του αντιγράφου ολοκληρώθηκε με επιτυχία,
19 με εμφάνιση μηνύματος στο τερματικό
20 if [ $? -eq 0 ]; then
    echo "Backup completed successfully: $BACKUP_NAME"
else
    echo "Backup failed"
fi

```

- **1.3 Δίνουμε δικαιώματα εκτέλεσης του αρχείου με την εντολή:**

```

1  sudo chmod +x /bin/labrat_backup_script.sh  sh

```

2. Ρύθμιση του Anacron

Αντί να χρησιμοποιήσουμε το cron για προγραμματισμένη εκτέλεση, θα χρησιμοποιήσουμε το anacron, το οποίο εξασφαλίζει ότι η εργασία θα εκτελεστεί ακόμα και αν ο υπολογιστής ήταν κλειστός κατά την προγραμματισμένη ώρα.

Για να το κάνουμε αυτό, πρέπει να προσθέσουμε την εργασία στο αρχείο `/etc/anacrontab`, το οποίο περιέχει τις ρυθμίσεις για το anacron.

- **2.1 Ανοίγουμε για επεξεργασία το αρχείο `/etc/anacrontab` με την επόμενη εντολή:**

```

1  sudo nano /etc/anacrontab  sh

```

- **2.2 Προσθέτουμε την παρακάτω γραμμή για να εκτελείται το script κάθε εβδομάδα, ακόμα κι αν ο υπολογιστής ήταν κλειστός:**

```

1  7      0      backup_weekly    /bin/labrat_backup_script.sh  sh
2

```

Επεξήγηση γραμμής `anacron`

- **7:** Αντιπροσωπεύει τη συχνότητα εκτέλεσης σε μέρες (7 ημέρες, δηλαδή εβδομαδιαία).
- **0:** Είναι η καθυστέρηση εκκίνησης (σε μέρες). Ορίζει ότι το `anacron` θα εκκινήσει την εργασία αμέσως μόλις ο υπολογιστής ανοίξει, αν η εργασία χάθηκε.
- **backup_weekly:** Είναι το όνομα της εργασίας. Μπορεί να είναι οποιοδήποτε όνομα, για να αναγνωρίσουμε την εργασία στο μέλλον.
- **/path/to/backup_script.sh:** Είναι η πλήρης διαδρομή προς το script μας.

3. Τρόπος λειτουργίας

- Το **anacron** εκτελεί την εργασία όταν ο υπολογιστής είναι ενεργός, ακόμα και αν το χρονικό διάστημα (π.χ., κάθε εβδομάδα) έχει παρέλθει. Αν ο υπολογιστής ήταν κλειστός την καθορισμένη ώρα, η εργασία θα εκτελεστεί μόλις ο υπολογιστής ανοιχτεί την επόμενη φορά.
- Εφόσον το `anacron` είναι ήδη ενσωματωμένο στο Ubuntu, δεν χρειάζεται να το εγκαταστήσουμε. Απλά το διαμορφώνουμε μέσω του αρχείου `anacrontab`.

Με τη χρήση του **anacron**, μπορούμε να εξασφαλίσουμε ότι τα αντίγραφα ασφαλείας θα δημιουργούνται ακόμη και αν ο υπολογιστής είναι κλειστός τη στιγμή που έχει προγραμματιστεί η εκτέλεση. Αυτό καθιστά το σύστημα πιο ευέλικτο και αξιόπιστο, καθώς το `anacron` θα αναλάβει την εκτέλεση μόλις το σύστημα είναι σε λειτουργία, χωρίς να χάνονται οι προγραμματισμένες εργασίες.

Εν κατακλείδι...

Το **Shell scripting** αποτελεί αναπόσπαστο εργαλείο για φοιτητές πληροφορικής, προσφέροντας δυνατότητες αυτοματοποίησης, βελτιστοποίησης διαδικασιών και εμβάθυνσης στη λειτουργία των συστημάτων. Η εκμάθησή του ανοίγει τον δρόμο για πιο αποδοτική διαχείριση συστημάτων και εξειδικευμένες εφαρμογές.

Βιβλιογραφία

1. Εισαγωγή στο *Shell Scripting* για Linux: https://linuxcommand.org/lc3_lts0010.php
2. Οδηγός *Bash* για Αρχάριους (*The Linux Documentation Project - TLDP*): <http://www.tldp.org/LDP/Bash-Beginners-Guide/html/>
3. Σελίδα εγχειριδίου *Anacron* (*Ubuntu Manpages*): <https://manpages.ubuntu.com/>

[manpages/jammy/man8/anacron.8.html](https://manpages.jammy/man8/anacron.8.html)

4. Cron - ArchWiki: <https://wiki.archlinux.org/title/Cron>

5. Εγχειρίδιο Αναφοράς GNU Bash: <https://www.gnu.org/savannah-checkouts/gnu/bash/manual/bash.html>

6. Χρήση του `tar` για τη Δημιουργία Αρχείου (GNU): https://www.gnu.org/software/tar/manual/html_node/Creating-Archives.html

7. `sed` - Επεξεργαστής Ροής GNU (GNU Stream Editor): <https://www.gnu.org/software/sed/manual/sed.html>

Server Installation

👤 Ευαγγελία Κατσιαντώνη 📅 10/28/24 🖥️ Linux 🔑 server, installation ⌚ Περίπου 19 λεπτά

Εισαγωγή

Η εγκατάσταση ενός server σε εικονικό περιβάλλον είναι ιδανική για εκπαιδευτικούς σκοπούς και δοκιμές, καθώς επιτρέπει τον πειραματισμό χωρίς να επηρεάζεται το κύριο λειτουργικό σύστημα.

Αυτός ο οδηγός θα σας καθοδηγήσει βήμα προς βήμα στην εγκατάσταση ενός Linux Server, σε μια εικονική μηχανή (VM) στο VirtualBox. Απευθύνεται σε σπουδαστές και τεχνικούς καλύπτοντας όλα τα βήματα, από τη δημιουργία της VM μέχρι τις βασικές ρυθμίσεις του server. Ο Server έχει όνομα (hostname) **DL-THESIS-SRV** και είναι παραμετροποιημένος σύμφωνα με τα στοιχεία που αναφέρονται στον πίνακα της παραγράφου Διαχείριση (Administration).

Σημείωση

Ο οδηγός είναι αυτός που χρησιμοποιήθηκε για τη δημιουργία της συσκευής **DL-Thesis-UTH.ova** που αναφέρεται στην Εισαγωγή.

Προαπαιτούμενα

Στις παρακάτω λίστες θα βρείτε όλα τις απαραίτητες πληροφορίες, απαιτήσεις στην ουσία, τόσο σε Software - Hardware, όσο και κάποιες βασικές γνώσεις ώστε να είναι πλήρως κατανοητός και υλοποιήσιμος αυτός ο οδηγός.

Υλικό

- Υπολογιστής με επεξεργαστή που υποστηρίζει Virtualization (VT-x/AMD-V).
- Τουλάχιστον 8GB RAM (προτείνεται 16GB για καλύτερη απόδοση).
- 50GB ελεύθερου χώρου στο δίσκο για το VM.

Λογισμικό

- VirtualBox , κατεβάστε και εγκαταστήστε την τελευταία έκδοση του VirtualBox από

την επίσημη ιστοσελίδα.

- Extension Pack του VirtualBox για υποστήριξη USB και άλλων λειτουργιών των εικονικών μηχανών.
- Εικόνα ISO του Ubuntu Server 24.04.2 LTS , κατεβάστε το ISO image του Ubuntu 24.04 LTS Server από την επίσημη ιστοσελίδα του Ubuntu.

Γνώσεις

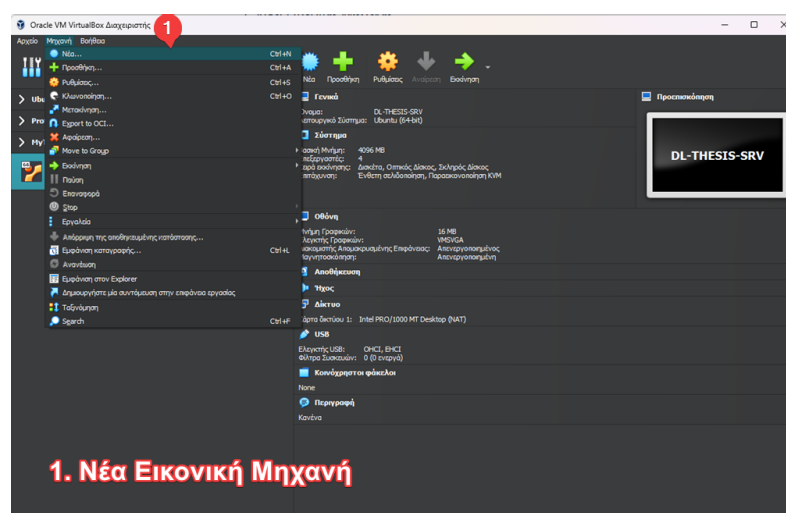
- Βασικές εντολές διαχείρισης Linux (Basic sysadmin commands).
- Κατανόηση δικτύων (IP, DNS, DHCP).

Δημιουργία εικονικής μηχανής

Αρχικά για να εγκαταστήσουμε το λειτουργικό Ubuntu 24.04.1 LTS σε κάποιο σύστημα που θα λειτουργεί ως Server πρέπει να δημιουργήσουμε το σύστημα αυτό. Για τη δημιουργία του επιλέγουμε τις τεχνολογίες εικονικοποίησης (αντί φυσικού συστήματος) με τη χρήση της εφαρμογής VirtualBox, έτσι ώστε να δημιουργήσουμε μια εικονική μηχανή σε ρόλο Linux Server.

1. Νέα Εικονική Μηχανή

Στο λειτουργικό σύστημα που χρησιμοποιείτε, ανοίξτε την εφαρμογή VirtualBox και από κύριο Μενού κάντε κλικ στο "Μηχανή -> New".



Εικ.1.1: Νέα Εικονική Μηχανή

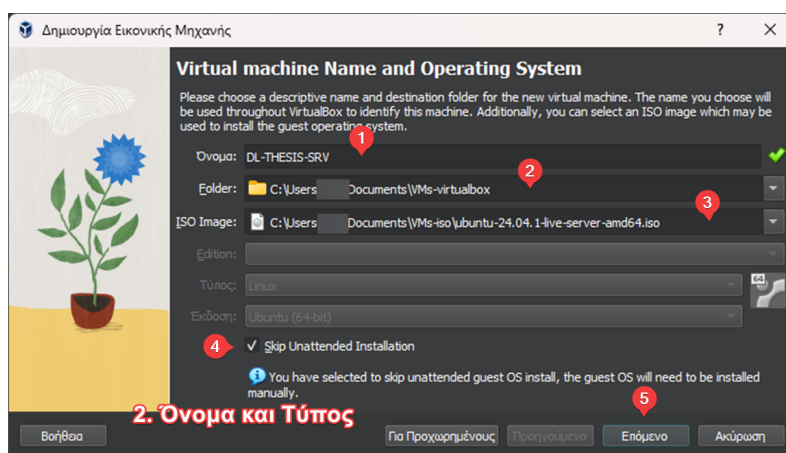
2. Όνομα και Τύπος

Συμπληρώστε στη φόρμα του αναδυόμενου παραθύρου τα παρακάτω στοιχεία:

- **1. Όνομα:** DL-THESIS-SRV
- **2. Folder:** επιλέξτε τον κατάλογο που θα δημιουργηθεί η εικονική μηχανή. (Από προεπιλογή η εφαρμογή το έχει συμπληρωμένο, προτείνεται να μην τον αλλάξετε).
- **3. ISO Image:** επιλέξτε τον κατάλογο που έχετε αποθηκευμένο το αρχείο ISO της εγκατάστασης που πρέπει να έχετε κατεβάσει σύμφωνα με τα προαπαιτούμενα.

Τα ακόλουθα τρία πεδία πρέπει να συμπληρωθούν αυτόματα μετά την επιλογή του αρχείου εγκατάστασης ISO

- **Edition:** -
- **Τύπος:** Linux
- **Έκδοση:** Ubuntu (64-bit)
- **4. Skip Unattended Installation:** Πριν συνεχίσετε επιλέξτε το για να παραλείψετε τη διαδικασία αυτόματης εγκατάστασης του λειτουργικού συστήματος, η οποία συνήθως απαιτεί προκαθορισμένες ρυθμίσεις και scripts που επιτρέπουν την εγκατάσταση χωρίς ανθρώπινη παρέμβαση.

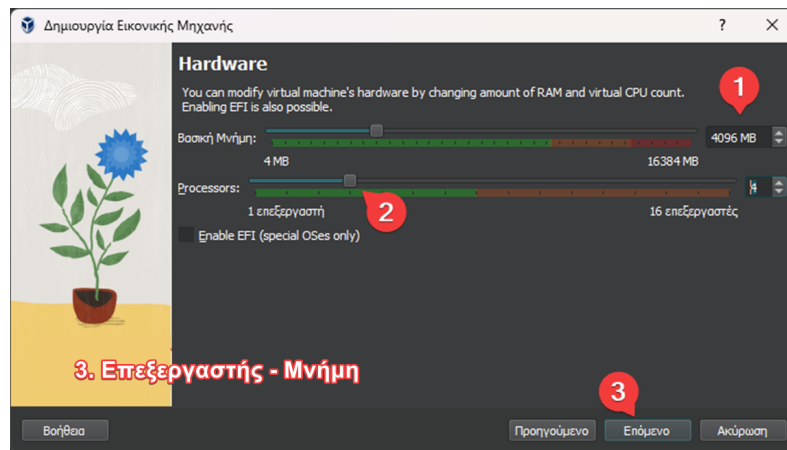


Εικ.1.2: Όνομα και Τύπος

3. Επεξεργαστής - Μνήμη

Ορίστε τα βασικά χαρακτηριστικά (Μνήμη - Επεξεργαστή), είτε πληκτρολογώντας τις κατάλληλες αριθμητικές τιμές στα πεδία δεξιά, είτε μετακινώντας τα slider στις προκαθορισμένες τιμές τους.

- **1. Βασική μνήμη:** τουλάχιστον 4096 MB (4GB) RAM
- **2. Processors:** τουλάχιστον 4 για τους πυρήνες στον επεξεργαστή
- **3. Συνεχίστε στο Επόμενο**



Εικ.1.3: Επεξεργαστής - Μνήμη

4. Σκληρός Δίσκος

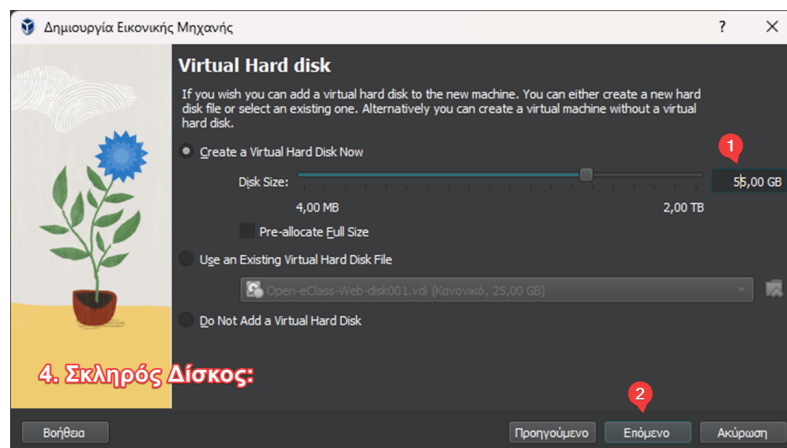
Ορίζουμε τη χωρητικότητα του δίσκου στον οποίο θα εγκατασταθεί και θα εκτελείται το λειτουργικό σύστημα.

- 1. Επιλέξτε **Create a virtual hard disk now**.
 - **Disk Size** Στο πεδίο πληκτρολογήστε την τιμή **55,00 GB**
 - Αφήστε τις άλλες προεπιλογές.

Important

- Βασικό για να μην δεσμεύεται άσκοπα χώρος στον υπολογιστή σας, μην επιλέγετε το **Pre-allocated Full Size**.
- Αντίθετα όταν ο χώρος στον δίσκο δεν είναι πρόβλημα ή οι εφαρμογές απαιτούν υψηλή απόδοση (π.χ., servers, databases, gaming) επιβάλλεται η επιλογή του.

- 2. Επόμενο για να συνεχιστεί η διαδικασία.

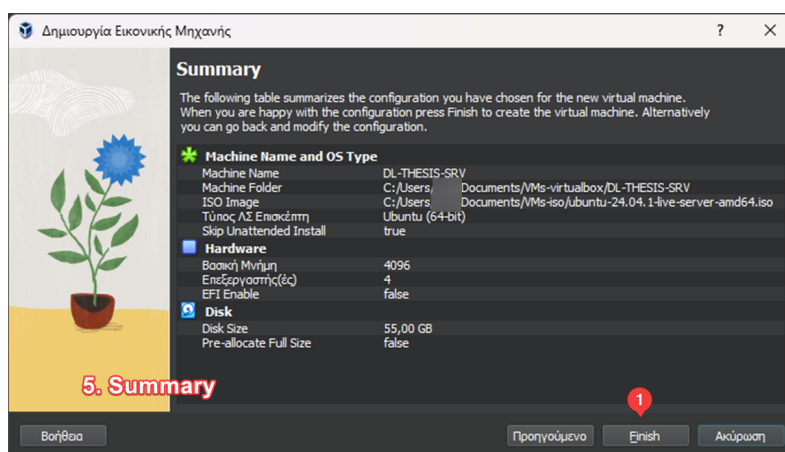


Εικ.1.4: Σκληρός Δίσκος

5. Προεπισκόπηση ρυθμίσεων

Προτού ξεκινήσετε την εγκατάσταση η καρτέλα Summary σας δίνει μια γρήγορη εικόνα των βασικών χαρακτηριστικών της εικονικής μηχανής, χωρίς να χρειάζεται να πλοηγηθείτε στις προηγούμενες για την επιβεβαίωση των ρυθμίσεων. Θα ακολουθηθείτε τη διαδικασία της πλοήγησης σε προηγούμενες εάν πρέπει να τροποποιήσετε κάποιο από τα στοιχεία.

- 1. Πατήστε **Finish** για να ολοκληρωθεί ο οδηγός και να δημιουργηθεί η εικονική μηχανή.



Εικ.1.4: Προεπισκόπηση ρυθμίσεων

Αναθεώρηση ρυθμίσεων

Μετά τη δημιουργία της εικονικής μηχανής στο αριστερό τμήμα του παραθύρου της εφαρμογής VirtualBox εμφανίζεται μια λίστα από όλες της μηχανές που έχουμε στον υπολογιστή μας. Υπάρχει επιλογή να τις εντάξουμε σε ομάδες για διευκόλυνση στη διαχείρισή τους.

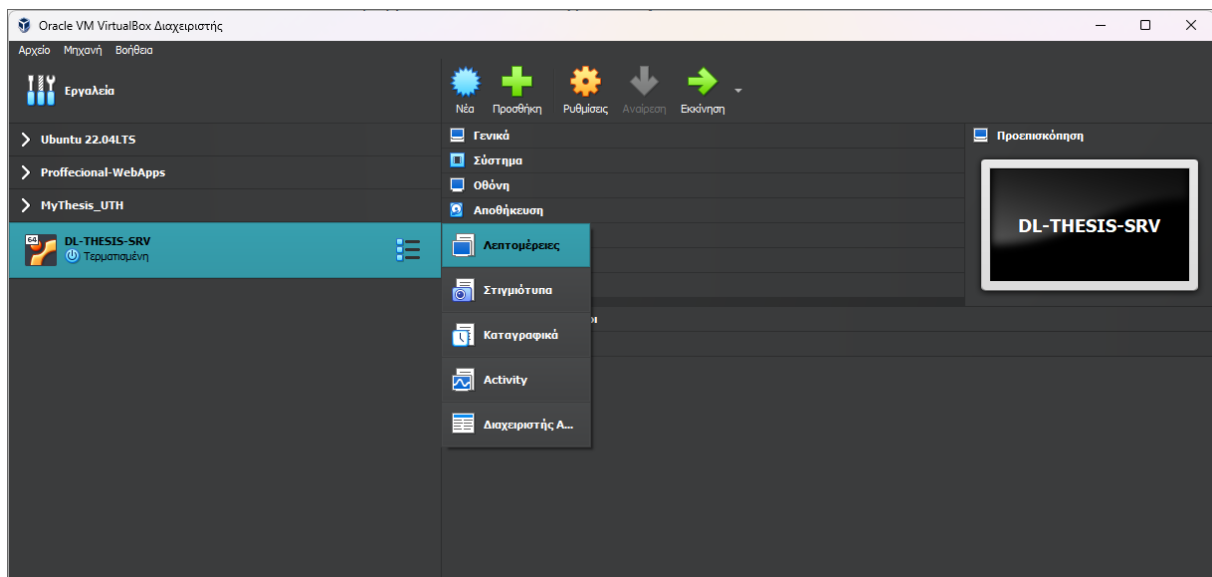
Επιλέγοντας (με μονό αριστερό κλικ) το VM που μόλις δημιουργήσατε, στο δεξιό τμήμα εμφανίζονται κάποιες πληροφορίες ανάλογα με το υποσύστημα που είναι επιλεγμένο στην περιοχή του εικονιδίου με τις τρεις τελείες και τις τρεις γραμμές. Οι διαθέσιμες επιλογές είναι **Λεπτομέρειες**, **Στιγμιότυπα**, **Καταγραφικά**, **Activity** και **Διαχειριστής Αρχείων**.

Για να δούμε και να αναθεωρήσουμε τις αρχικές μας ρυθμίσεις επιλέγουμε **Λεπτομέρειες** και στη συνέχεια στη δεξιά καρτέλα κάνουμε διπλό αριστερό κλικ σε όποια από τα χαρακτηριστικά θέλουμε να δούμε αναλυτικά ή να τροποποιήσουμε. Η Συλλογή -1-

παρέχει μια πιο κατατοπιστική σειρά εικόνων.

Στον οδηγό [How to Use VirtualBox](#) μάθετε τι είναι το Oracle VirtualBox, πώς να το εγκαταστήσετε στον υπολογιστή σας και πώς να το εκτελέσετε.

Επιλογή Εργαλείου

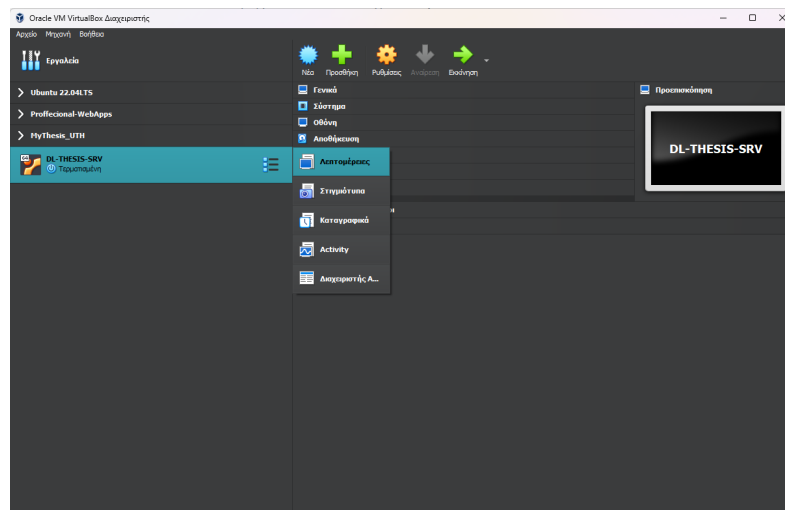


Συλλογή-1: Αναθεώρηση ρυθμίσεων

Εγκατάσταση

1. Εκκίνηση VM

Επιλέξτε το DL-THESIS-SRV και κάντε κλικ στον πράσινο διακόπτη "Εκκίνηση" στο πάνω μέρος του παραθύρου.



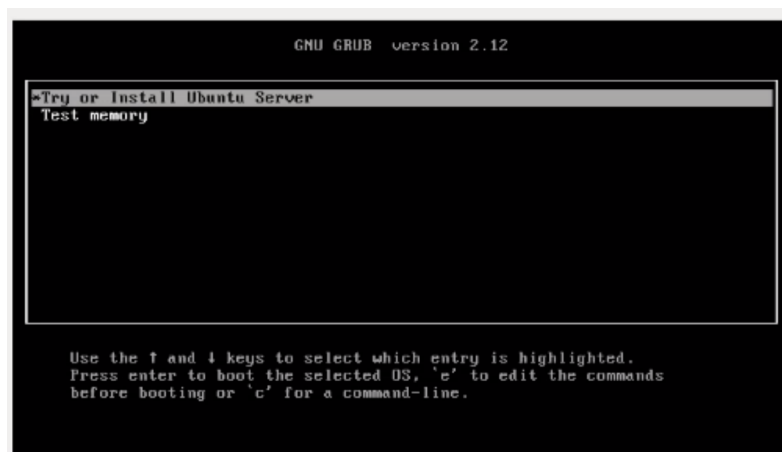
Εικ.-6

2. Επιλογή *boot entry*

Στην οθόνη **GRUB menu entries** εάν δεν είναι επιλεγμένο το **Try or Install Ubuntu Server** με τα βέλη (πάνω / κάτω) του πληκτρολογίου επιλέξτε το και στη συνέχεια πατήστε **Enter**. Σε περίπτωση που είναι ήδη έτοιμη είτε περιμένετε 30 δευτερόλεπτα παρατηρώντας το χρονόμετρο στο κάτω μέρος της οθόνης του τερματικού, είτε πατάτε άμεσα [Enter] για να ξεκινήσει η διαδικασία της εγκατάστασης.

Στην οθόνη εμφανίζονται να "τρέχουν" γραμμές, οι οποίες μας παρέχουν πληροφορίες για το τι εκτελείται. Ειδικότερα έχουμε:

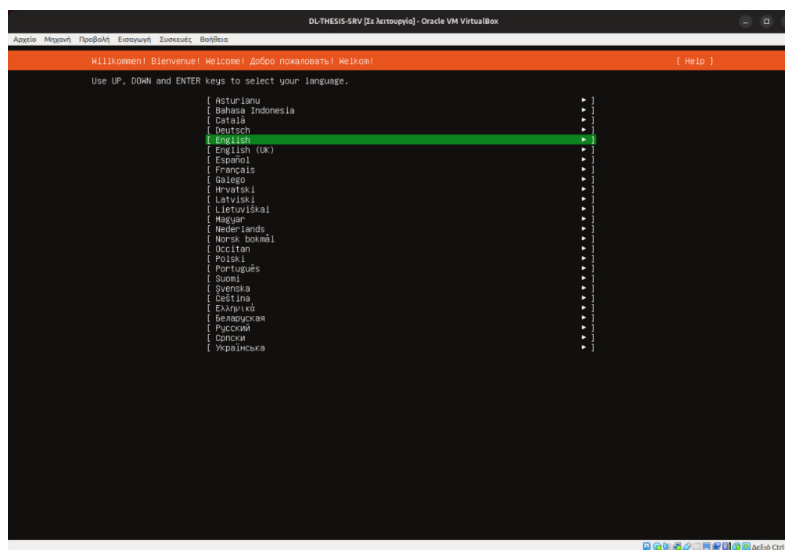
- **Φόρτωση του πυρήνα (kernel):** Το πρώτο βήμα είναι η φόρτωση του Linux kernel στη μνήμη.
- **Εκκίνηση υπηρεσιών (services):** Μετά την φόρτωση του πυρήνα, ξεκινούν διάφορες υπηρεσίες του συστήματος, όπως ο διαχειριστής δικτύου, ο διαχειριστής αρχείων και άλλες. Κάθε γραμμή που βλέπετε συνήθως αντιστοιχεί σε μία υπηρεσία που ξεκινά.
- **Έλεγχος υλικού:** Το σύστημα ελέγχει το υλικό του υπολογιστή, όπως τον επεξεργαστή, τη μνήμη και τους δίσκους.
- **Φόρτωση drivers:** Το σύστημα φορτώνει τα απαραίτητα drivers για το υλικό.
- **Προετοιμασία του περιβάλλοντος χρήστη:** Το σύστημα προετοιμάζει το περιβάλλον χρήστη (γραφικό ή γραμμής εντολών) για να μπορέσετε να συνδεθείτε.



Εικ.-7

3. Γλώσσα και Πληκτρολόγιο

Ακολουθείτε τις οδηγίες που εμφανίζονται, στην οθόνη καλωσορίσματος για να επιλέξετε με **Enter** τη γλώσσα και τη διάταξη πληκτρολογίου. Προεπιλογή είναι τα Αγγλικά οπότε και προτείνεται να συνεχίσετε.



Εικ.-8

4. Ενημέρωση του προγράμματος εγκατάστασης

Ανάλογα με ποια έκδοση του ISO έχετε κατεβάσει, κατά τη διάρκεια της εγκατάστασης, το installer μπορεί να ελέγξει για ενημερώσεις. Αν βρεθεί νεότερη έκδοση, σας δίνεται η επιλογή να την κατεβάσετε και να την χρησιμοποιήσετε. Αυτό γίνεται για να εξασφαλιστεί ότι η εγκατάσταση θα γίνει με την πιο πρόσφατη και βελτιωμένη έκδοση του installer, η οποία μπορεί να περιλαμβάνει:

- **Διορθώσεις σφαλμάτων (bug fixes):** Η νέα έκδοση μπορεί να διορθώνει σφάλματα

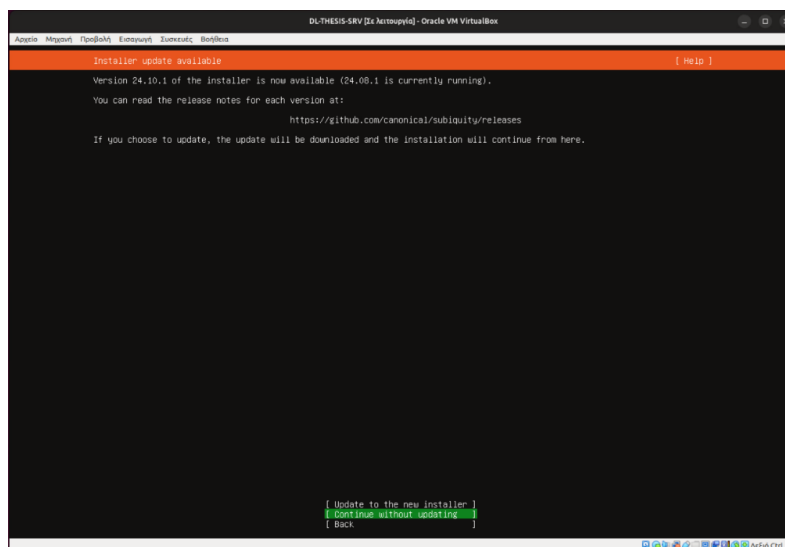
που υπήρχαν σε προηγούμενες εκδόσεις.

- **Νέα χαρακτηριστικά:** Μπορεί να προσθέτει νέες λειτουργίες ή βελτιώσεις.
- **Καλύτερη συμβατότητα:** Μπορεί να βελτιώνει τη συμβατότητα με νεότερο υλικό.

Εσείς απλά επιλέξτε με τον γνωστό τρόπο (πάνω/κάτω βέλη πληκτρολογίου και Enter) μια από τις επόμενες διαθέσιμες επιλογές.

- Update to the new installer
- Continue without updating
- Back

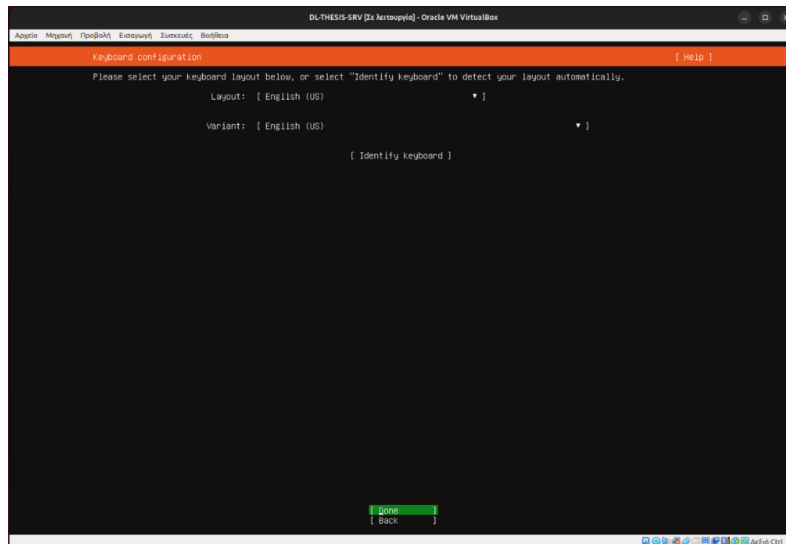
Προσωπικά **πάντοτε** επιλέγω να ενημερώσω στην νεότερη έκδοση τον εγκαταστάτη και να συνεχίσω.



Εικ.-9

5. Διαμόρφωση διάταξης πληκτρολογίου

Ολοκληρώνοντας την αναβάθμιση εμφανίζεται η οθόνη "Keyboard Configuration" του Ubuntu Server (και άλλων διανομών Linux) που σας επιτρέπει να ορίσετε τη διάταξη (layout) του πληκτρολογίου που θα χρησιμοποιηθεί στο σύστημα. Αυτή η ρύθμιση είναι κρίσιμη για την σωστή εισαγωγή χαρακτήρων και συμβόλων, καθώς διαφορετικές γλώσσες και περιοχές χρησιμοποιούν διαφορετικές διατάξεις πληκτρολογίου. Σε περιβάλλοντα server, όπου η πρόσβαση γίνεται κυρίως μέσω γραμμής εντολών, η σωστή ρύθμιση πληκτρολογίου είναι απαραίτητη για την εκτέλεση εντολών και την διαχείριση του συστήματος.



Εικ.-10

Επιλέξτε English, και αφού ελεγξετε τη διάταξη, πατήστε **Done** να συνεχίσει ο installer στον τύπο της εγκατάστασης.

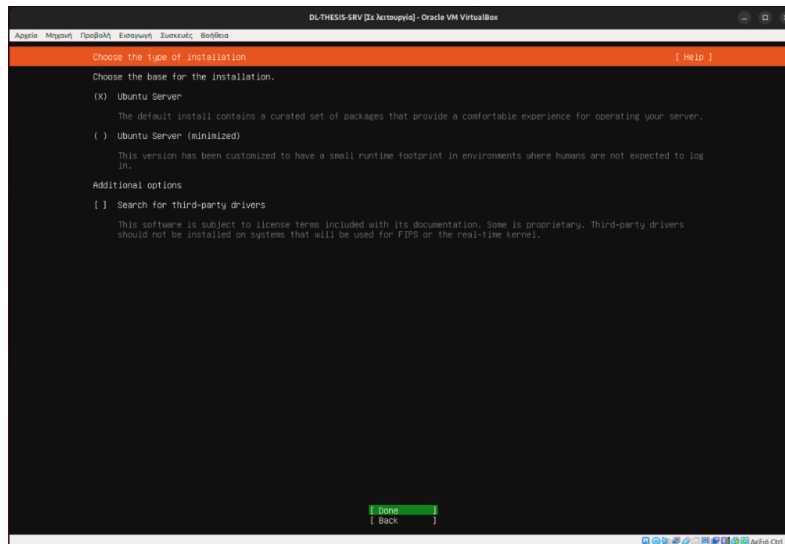
6. Τύπος Εγκατάστασης

Αυτή η επιλογή σας επιτρέπει να καθορίσετε ποια **πακέτα λογισμικού** θα εγκατασταθούν αρχικά στο σύστημά σας.

☒ **Ubuntu server:** Εγκαθιστά ένα πιο πλήρες σύνολο πακέτων, συμπεριλαμβανομένων εργαλείων για διαχείριση server και άλλων βασικών εφαρμογών. Δεν περιλαμβάνει γραφικό περιβάλλον (GUI) και είναι προεπιλογή του Ubuntu.

☐ **Ubuntu server (minimal installation):** Εγκαθιστά ένα ελάχιστο σύνολο πακέτων, ιδανικό για servers όπου απαιτείται εξοικονόμηση πόρων. Επίσης δεν περιλαμβάνει γραφικό περιβάλλον (GUI).

☒ **Search for third-party drivers:** Κατά την εγκατάσταση του Ubuntu Server επιτρέπει την αναζήτηση και εγκατάσταση drivers (οδηγών) για υλικό που δεν υποστηρίζεται πλήρως από τους open-source drivers που περιλαμβάνονται στην διανομή. Είναι **σημαντικό** να το επιλέξετε γιατί ορισμένες συσκευές, όπως κάρτες γραφικών, κάρτες δικτύου, ή άλλο υλικό, απαιτούν ιδιόκτητους drivers για να λειτουργήσουν σωστά. Αυτοί οι drivers συνήθως παρέχονται από τους κατασκευαστές του υλικού.



Εικ.-11

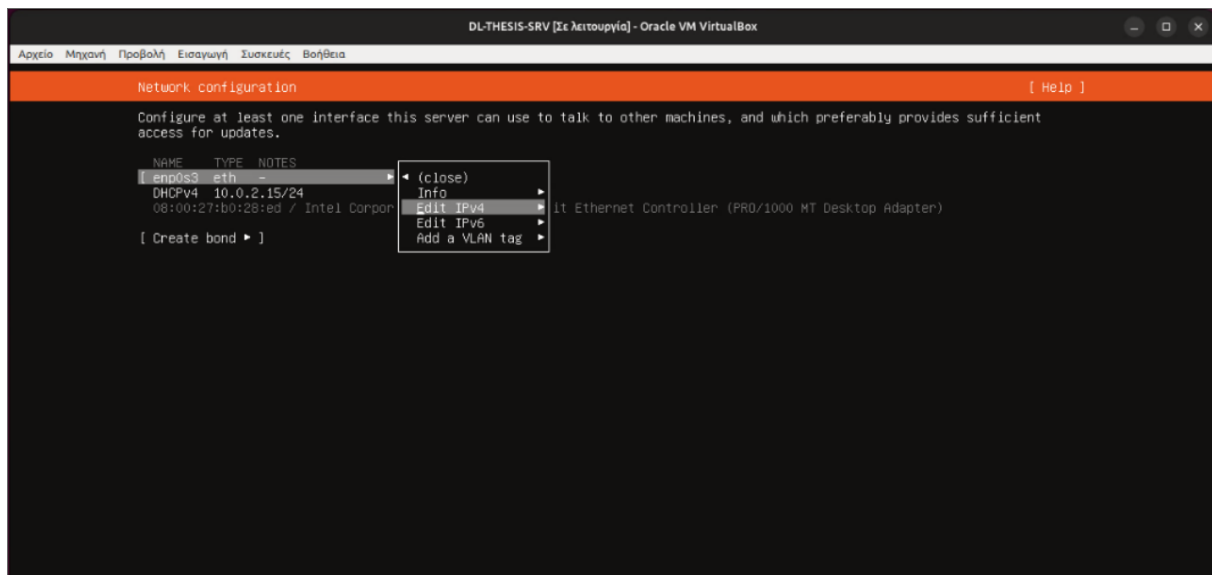
7. Διαμόρφωση Δικτύου

Η διαμόρφωση δικτύου κατά την εγκατάσταση ενός server είναι κρίσιμη για την συνδεσιμότητα του συστήματος. Οι διαθέσιμες επιλογές περιλαμβάνουν την **αυτόματη λήψη IP διεύθυνσης μέσω DHCP**, την **χειροκίνητη ρύθμιση μιας σταθερής IP διεύθυνσης**, και πιο προηγμένες τεχνικές όπως το **"bonding" πολλών κάρτων δικτύου**. Η επιλογή της κατάλληλης μεθόδου εξαρτάται από τις ανάγκες και την τοπολογία του δικτύου.

Αναλυτικά:

- ☐ **DHCP:** Η πιο απλή επιλογή. Ο server θα λάβει αυτόματα μια IP διεύθυνση, netmask, gateway, και DNS servers από έναν DHCP server στο δίκτυό σας (συνήθως ο router).
- ☒ **Static IP:** Σε αυτή την περίπτωση, πρέπει να ορίσετε χειροκίνητα την IP διεύθυνση, netmask, gateway, και DNS servers. Αυτή η επιλογή είναι χρήσιμη όταν θέλετε ο server να έχει μια σταθερή IP διεύθυνση.
- ☐ **Bonding:** Η σύνδεση πολλών κάρτων δικτύου για να δημιουργήσετε ένα "virtual" interface με αυξημένη ταχύτητα και αξιοπιστία.

Ορισμός Σταθερής IP

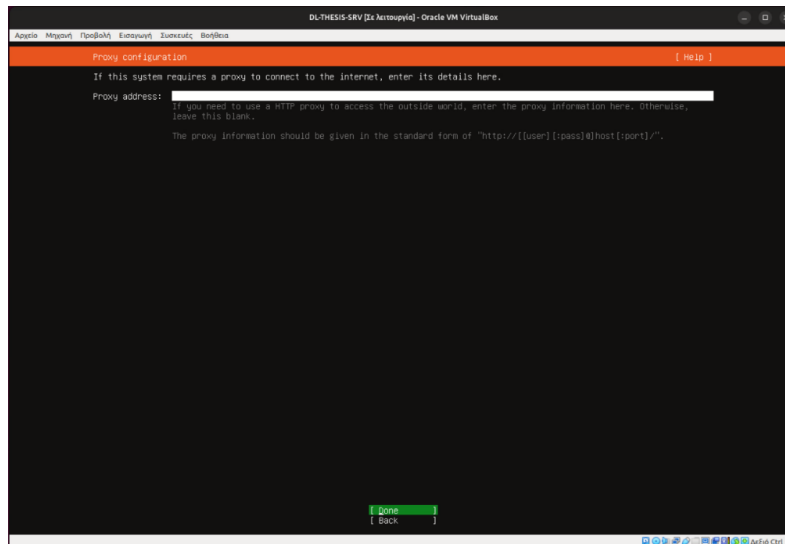


Συλλογή-2: Διαμόρφωση Δικτύου

8. Διαμόρφωση Proxy

Η ρύθμιση proxy κατά την εγκατάσταση ενός λειτουργικού συστήματος είναι απαραίτητη σε περιβάλλοντα όπου η πρόσβαση στο internet γίνεται μέσω proxy server. Ο proxy server λειτουργεί ως ενδιάμεσος μεταξύ του συστήματος και του internet, και η σωστή ρύθμισή του επιτρέπει στον installer να κατεβάσει τα απαραίτητα πακέτα. Σε αυτήν την περίπτωση, ο χρήστης πρέπει να παρέχει τη διεύθυνση και το port του proxy server, καθώς και τα διαπιστευτήρια εάν απαιτούνται.

Αφήστε τη ρύθμιση κενή



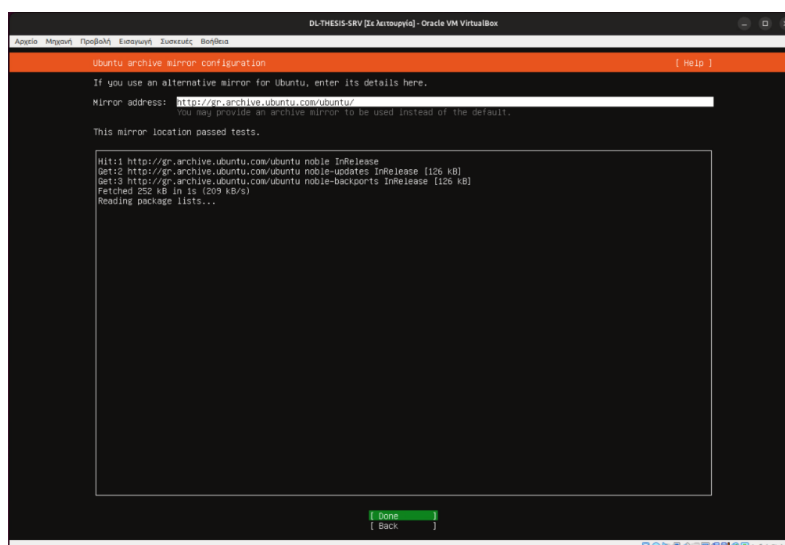
Εικ.-18

9. Επιλογή Καθρέφτη Αποθετηρίων

Η επιλογή καθρέφτη (mirror) για τα αποθετήρια λογισμικού κατά την εγκατάσταση ενός λειτουργικού συστήματος επιτρέπει την λήψη των απαραίτητων πακέτων από μια τοποθεσία που είναι γεωγραφικά πιο κοντά ή πιο αξιόπιστη. Η επιλογή του κατάλληλου καθρέφτη μπορεί να επηρεάσει σημαντικά την ταχύτητα και την αξιοπιστία της εγκατάστασης.

- **Αυτόματη επιλογή:** Ο installer συνήθως προτείνει έναν κατάλληλο καθρέφτη αυτόματα.

Περιμένετε προτού πατήσετε **Done** μέχρι να δείτε στο τερματικό τη φράση **Reading package lists...**



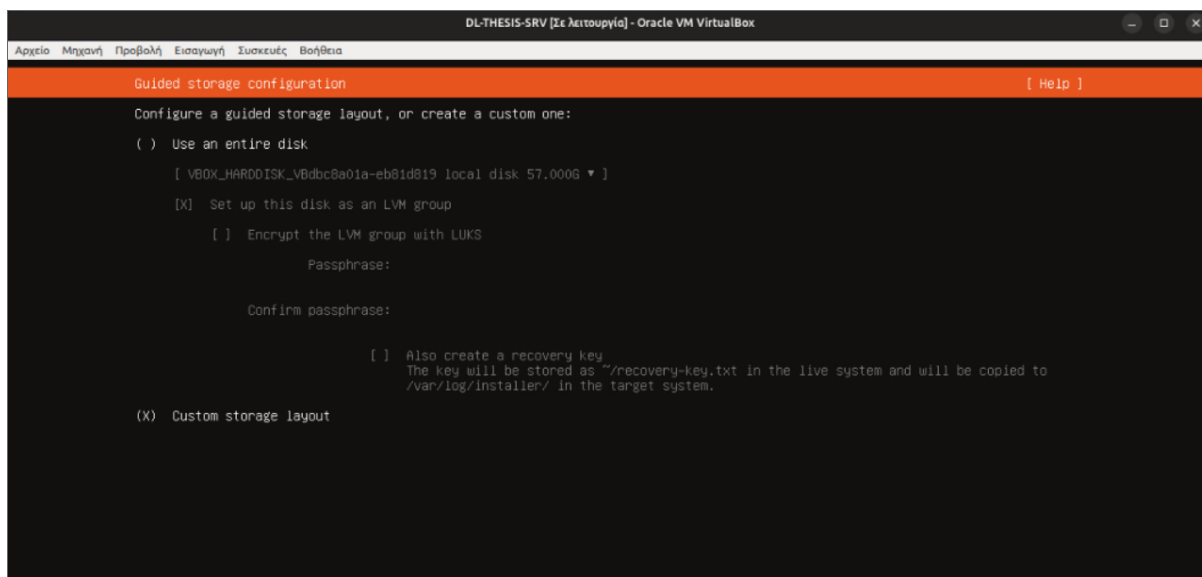
Εικ.-18

10. Καθοδηγούμενη διαμόρφωση αποθήκευσης

Η οθόνη "Guided Storage Configuration" κατά την εγκατάσταση του Ubuntu Server προσφέρει ένα εύρος επιλογών για την διαμόρφωση των partitions. Από την απλή χρήση ενός ολόκληρου δίσκου έως την χρήση LVM και κρυπτογράφησης, ο χρήστης μπορεί να επιλέξει την βέλτιστη λύση ανάλογα με τις απαιτήσεις του συστήματος. Για απλές εγκαταστάσεις, η επιλογή "Use An Entire Disk" είναι συνήθως επαρκής. Για συστήματα που απαιτούν δυναμική διαχείριση χώρου ή αυξημένη ασφάλεια, η χρήση LVM ή κρυπτογραφημένου LVM αντίστοιχα, είναι ενδεδειγμένη. Η επιλογή "Something else" προσφέρει μέγιστη ευελιξία για προχωρημένους χρήστες, αλλά απαιτεί βαθύτερη κατανόηση των διαδικασιών partitioning.

- ☐ **Use An Entire Disk:** Χρησιμοποιεί ολόκληρο τον δίσκο για την εγκατάσταση του Ubuntu. Αυτή είναι η απλούστερη επιλογή, ιδανική για συστήματα με έναν μόνο δίσκο. Επιλέγετε τον δίσκο και το σύστημα δημιουργεί αυτόματα τα απαραίτητα partitions.
- ☒ **Set up this disk as an LVM group (Logical Volume Management):** Αν επιλέξετε LVM, ο δίσκος σας θα διαμορφωθεί ως LVM. Το LVM επιτρέπει την δυναμική διαχείριση του χώρου στο δίσκο. Μπορείτε να αλλάξετε το μέγεθος των partitions ακόμα και μετά την εγκατάσταση. Συνιστάται η επιλογή της.
- ☐ **Use An Entire Disk And Set Up Encrypted LVM:** Αυτή η επιλογή συνδυάζει τα πλεονεκτήματα του LVM με την κρυπτογράφηση. Όλα τα δεδομένα στον δίσκο θα κρυπτογραφηθούν, προσφέροντας αυξημένη ασφάλεια.
- ☒ **Custom storage layout:** Αυτή η επιλογή την οποία ακολουθούμε και προτείνουμε στον οδηγό σας επιτρέπει να διαμορφώσετε τα partitions χειροκίνητα. Είναι η πιο ευέλικτη επιλογή, **αλλά απαιτεί περισσότερες τεχνικές γνώσεις**. Επιλέξτε αυτήν την επιλογή αν θέλετε να δημιουργήσετε προσαρμοσμένα partitions, να χρησιμοποιήσετε διαφορετικά file systems, ή να ρυθμίσετε RAID.

Ρυθμίσεις Διαμόρφωσης Δίσκου



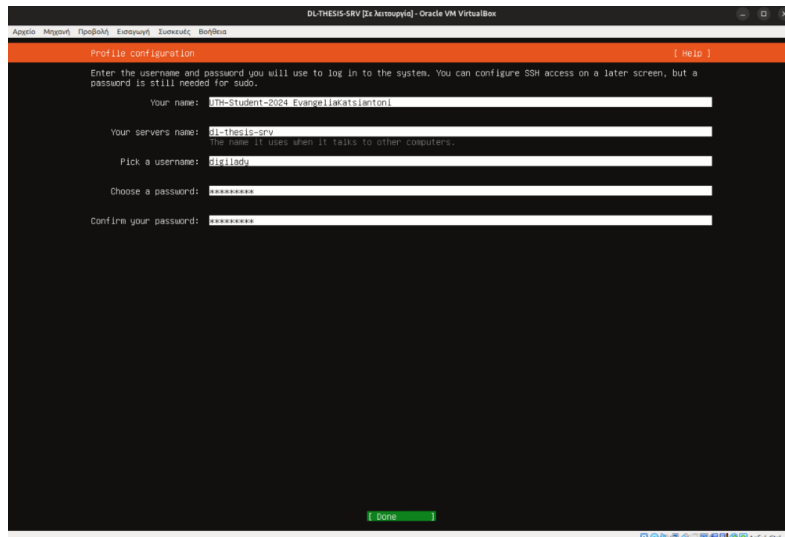
Συλλογή-3-

11. Διαμόρφωση προφίλ χρήστη

Η οθόνη "Profile Setup" κατά την εγκατάσταση του Ubuntu Server επιτρέπει την δημιουργία του αρχικού χρήστη και την ρύθμιση των βασικών στοιχείων του λογαριασμού. Κατά την διαδικασία αυτή, καθορίζεται το πλήρες όνομα του χρήστη, το όνομα του διακομιστή (hostname), καθώς και τα διαπιστευτήρια σύνδεσης, δηλαδή το όνομα χρήστη και ο κωδικός πρόσβασης. Συνιστάται ιδιαίτερα η χρήση ισχυρού κωδικού πρόσβασης, ο οποίος συμπεριλαμβάνει μικρά και κεφαλαία γράμματα, αριθμούς και σύμβολα, για την μέγιστη ασφάλεια του συστήματος.

Τα πεδία ρυθμίσεων είναι κατά σειρά τα:

- **Your name:** Το πλήρες όνομά σας. (έχει ρυθμιστεί στο **UTH-Student-2024 EvangeliaKatsiantoni**)
- **Your server's name:** Το όνομα του server (hostname). (έχει ρυθμιστεί στο **dl-thesis-srv**)
- **Pick a username:** Το όνομα χρήστη για σύνδεση. (έχει ρυθμιστεί στο **digilady**)
- **Choose a password:** Ένας ισχυρός κωδικός πρόσβασης. (έχει ρυθμιστεί στο **3edc+^YHN**)
- **Confirm your password:** Επιβεβαίωση του κωδικού πρόσβασης. (έχει ρυθμιστεί στο **3edc+^YHN**)



Εικ.-27

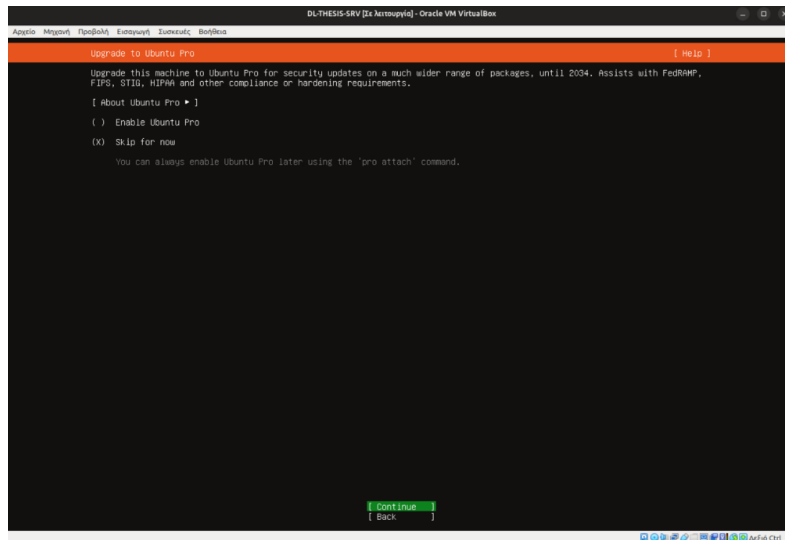
12. Upgrade to Ubuntu Pro

Μετά την διαμόρφωση του προφίλ χρήστη, η διαδικασία εγκατάστασης του Ubuntu Server παρέχει την δυνατότητα ενεργοποίησης της συνδρομητικής υπηρεσίας Ubuntu Pro. Η υπηρεσία αυτή προσφέρει μια σειρά από επιπλέον χαρακτηριστικά και υποστήριξη για επιχειρήσεις και οργανισμούς, όπως επεκτεταμένη υποστήριξη ασφαλείας, livepatching του πυρήνα και εργαλεία διαχείρισης πολλαπλών servers. Οι χρήστες έχουν την επιλογή να:

- ☐ **Enable Ubuntu Pro:** Ενεργοποιήσουν το Ubuntu Pro κατά την εγκατάσταση.
- ☒ **Skip for now:** Παραλείψουν και να προχωρήσουν στην τελική διαμόρφωση του συστήματος, που είναι και η επιλογή στο οδηγό.

Οι επιπλέον λειτουργίες που προσφέρει η υπηρεσία Ubuntu Pro:

- **Extended Security Maintenance (ESM):** Παρέχει ενημερώσεις ασφαλείας για ένα μεγαλύτερο χρονικό διάστημα από την standard υποστήριξη του Ubuntu.
- **Kernel Livepatching:** Επιτρέπει την εφαρμογή ενημερώσεων πυρήνα χωρίς επανεκκίνηση του συστήματος, βελτιώνοντας τη διαθεσιμότητα, ειδικά σε συστήματα που η λειτουργία τους είναι 24/7.
- **FIPS 140-2 certified modules:** Συμβατότητα με το πρότυπο ασφάλειας FIPS 140-2.
- **Landscape management tool:** Εργαλείο για τη διαχείριση πολλών Ubuntu servers.
- **Υποστήριξη για περισσότερα πακέτα λογισμικού:** Πρόσβαση σε ένα μεγαλύτερο αριθμό πακέτων λογισμικού από τα αποθετήρια του Ubuntu Pro.



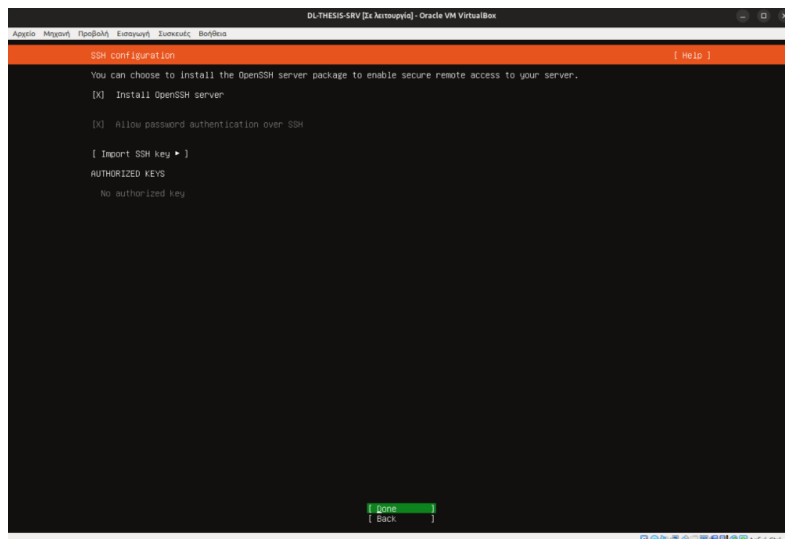
Εικ.-28

13. Εγκατάσταση - Διαμόρφωση SSH

Η οθόνη "SSH setup" κατά την εγκατάσταση του Ubuntu Server προσφέρει την δυνατότητα εγκατάστασης και βασικής ρύθμισης του OpenSSH server. Ο OpenSSH server είναι ένα κρίσιμο στοιχείο για την ασφαλή απομακρυσμένη διαχείριση ενός server, επιτρέποντας την σύνδεση μέσω SSH. Η εγκατάστασή του κατά την διάρκεια της εγκατάστασης του λειτουργικού συστήματος απλοποιεί την διαδικασία και συνιστάται για τους περισσότερους servers. Επιπλέον, η δυνατότητα εισαγωγής ενός υπάρχοντος SSH key προσφέρει αυξημένη ασφάλεια, καθιστώντας την απομακρυσμένη πρόσβαση πιο εύκολα προσβάσιμη και αποδοτική.

Οι επιλογές που εμφανίζονται είναι:

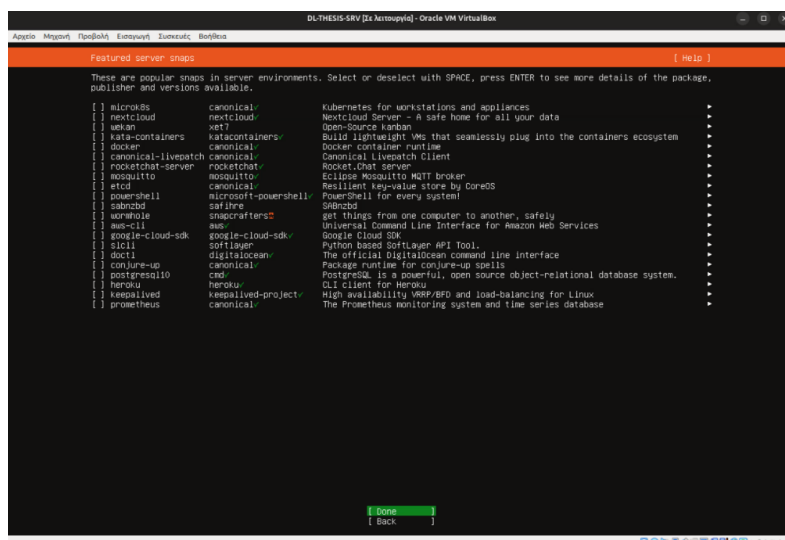
- ☒ **Install OpenSSH server:** Επιλέγοντας αυτό, θα εγκατασταθεί ο OpenSSH server στον υπολογιστή σας. Συνιστάται να επιλέξετε αυτήν την επιλογή, ειδικά αν σκοπεύετε να διαχειρίζεστε τον server από απόσταση, όπως χαρακτηριστικά αναφέρεται στο οδηγό.
- ☐ **Import SSH identity:** Μπορείτε να εισάγετε ένα υπάρχον SSH key για αυθεντικοποίηση. Αυτό σας επιτρέπει να συνδέεστε στον server χωρίς να χρειάζεται να πληκτρολογείτε τον κωδικό πρόσβασης κάθε φορά. Εάν δεν έχετε ήδη ένα SSH key, μπορείτε να το δημιουργήσετε αργότερα.



Εικ.-29

14. Featured Server Snaps

Η οθόνη "Featured Server Snaps" στην διαδικασία εγκατάστασης του Ubuntu Server προσφέρει μια επιλογή από δημοφιλή πακέτα λογισμικού σε μορφή snap. Τα snaps αποτελούν ένα σύγχρονο σύστημα διαχείρισης λογισμικού που απλοποιεί την εγκατάσταση και την ενημέρωση εφαρμογών, καθώς περιλαμβάνουν όλες τις απαραίτητες εξαρτήσεις. Η επιλογή εγκατάστασης κατά την διάρκεια της εγκατάστασης του λειτουργικού συστήματος διευκολύνει την αρχική ρύθμιση του server. Ωστόσο, έχετε την επιλογή να παραλείψετε την εγκατάσταση των snaps αυτών και να τα εγκαταστήσετε αργότερα, ανάλογα με τις ανάγκες σας. Στην εγκατάσταση του διακομιστή **DL-THESIS-SRV**, δεν επιλέχτηκε κανένα πακέτο.



Εικ.-30

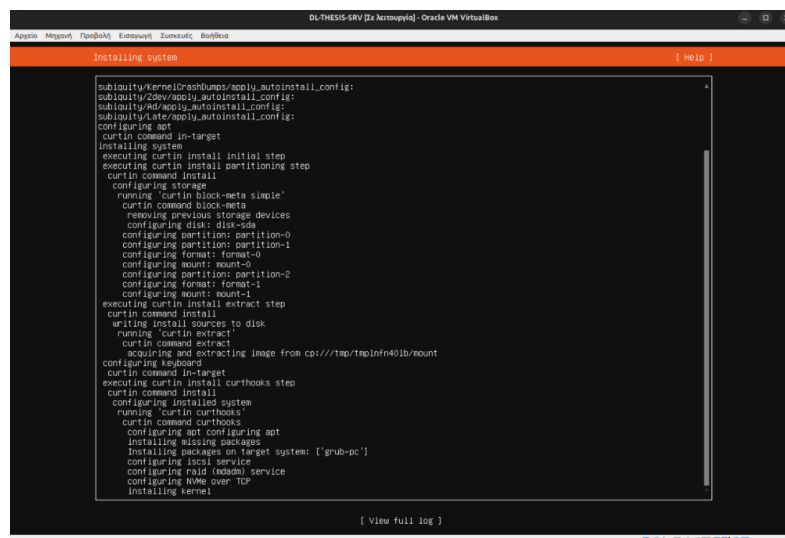
15. Εγκατάσταση συστήματος

Το τελικό στάδιο της εγκατάστασης του Ubuntu Server, "Installing system", είναι η φάση όπου το σύστημα αντιγράφει τα αρχεία στο δίσκο, εγκαθιστά το λειτουργικό σύστημα και ρυθμίζει τις βασικές υπηρεσίες. Κατά τη διάρκεια αυτής της φάσης, θα δείτε μια σειρά από μηνύματα που δείχνουν την πρόοδο της εγκατάστασης, όπως:

- Copying files...
- Installing packages...
- Configuring services...
- Finishing the installation...

Τι συμβαίνει όμως στο παρασκήνιο;

- **Εγκατάσταση του bootloader (GRUB):** Ο GRUB είναι ο bootloader που σας επιτρέπει να επιλέξετε ποιο λειτουργικό σύστημα να εκκινήσετε.
- **Ρύθμιση δικτύου:** Εάν έχουν οριστεί οι ρυθμίσεις δικτύου κατά την εγκατάσταση, το σύστημα θα τις ρυθμίσει σε αυτό το στάδιο.
- **Δημιουργία χρήστη:** Δημιουργεί τον χρήστη που ορίσατε στο "Profile Setup".
- **Εγκατάσταση των επιλεγμένων πακέτων:** Εγκαθιστά τα πακέτα που επιλέξατε κατά την εγκατάσταση (π.χ. OpenSSH server, featured server snaps).



```
subiquity/kernelDrashDumps/apply_autoinstall_config:
subiquity/2dev/apply_autoinstall_config:
subiquity/4d/apply_autoinstall_config:
subiquity/late/apply_autoinstall_config:
Configuring apt
curtin command in-target
Installing system
executing curtin install initial step
executing curtin install partitioning step
curtin command install
configuring storage
running 'curtin block-meta simple'
curtin command block-meta
removing previous storage devices
configuring disk: disk-sda
configuring partition: partition-0
configuring partition: partition-1
configuring format: format-0
configuring mount: mount-0
configuring partition: partition-2
configuring format: format-1
configuring mount: mount-1
executing curtin install extract step
curtin command install
writing install sources to disk
running 'curtin extract'
curtin command extract
acquiring and extracting image from cpi:///tpm/tpm.inf401b/mount
configuring keyboard
curtin command in-target
executing curtin install curthooks step
curtin command install
configuring installed system
running 'curtin curthooks'
curtin command curthooks
configuring apt configuring apt
Installing missing packages
Installing packages on target system: ['grub-pc']
configuring lsscsi service
configuring raid (mdadm) service
configuring kNMe over TCP
Installing kernel
```

Εικ.-31

16. Installation complete

Η οθόνη "Installation complete" σηματοδοτεί την επιτυχή ολοκλήρωση της εγκατάστασης του Ubuntu Server. Σε αυτό το στάδιο, έχετε τη δυνατότητα προβολής του πλήρους log της εγκατάστασης, ένα χρήσιμο εργαλείο για τους διαχειριστές συστημάτων που επιθυμούν να εξετάσουν λεπτομερώς την διαδικασία εγκατάστασης και να εντοπίσουν τυχόν προβλήματα. Αφού επιλέξετε την επανεκκίνηση του συστήματος ("Reboot now"), ο Ubuntu Server είναι πλέον έτοιμος προς χρήση, με όλες τις

- ☐ View full log
- ☒ Reboot now



1. Ενημέρωση Συστήματος

```
1 sudo apt update
2 sudo apt upgrade -y
```

2. Ρύθμιση Firewall (UFW)

```
1 sudo ufw enable
2 sudo ufw allow ssh
3 sudo ufw status
```

3. Ρύθμιση Χρονικής Ζώνης

```
1 sudo timedatectl set-timezone Europe/Athens sh
2 timedatectl status
```

Διαχείριση

Πίνακας με απαραίτητες ομαδοποιημένες πληροφορίες διαχείρισης του Linux Server (LSRV), ο οποίος είναι η βάση της συσκευής DL-THESIS-SRV.ova

System

- **ServerIP:** 192.168.1.146
- **ServerName** (Hostname): dl-thesis-srv (διαμορφώνεται στην εγκατάσταση)
- **Timezone:** Europe/Athens

Users

- **root** ο βασικός χρήστης της εγκατάστασης και ελεύθερης πρόσβασης στο terminal / console
- ***digilady* (*3edc+^YHN*)**, χρήστης με διαχειριστικά δικαιώματα (group:sudo), με δικό του φάκελο στο home

Για να δω τους χρήστες στο σύστημα μέσω τερματικού

- **1. Μόνο ονόματα χρηστών:**

```
1 cat /etc/passwd | cut -d: -f1 sh
```

- **2. Πλήρη στοιχεία των χρηστών:**

```
1 cat /etc/passwd sh
```

SSH

Ενεργοποίηση της πρόσβασης μέσω **SSH** στην προεπιλεγμένη πόρτα σύνδεσης (Port 22), χωρίς παραμετροποιήσεις στο αρχείο ρυθμίσεων **sshd_config**

Apps

Προεγκατεστημένες

Εφαρμογές οι οποίες εγκαταστάθηκαν κατά τη διαδικασία της εγκατάστασης του λειτουργικού συστήματος Ubuntu Server 24.04.1 LTS.

ΑΑ	Εφαρμογή	Χρήση	Πρόσβαση	Παρατηρήσεις
1.	openSSH	Remote Server CLI Administration		terminal

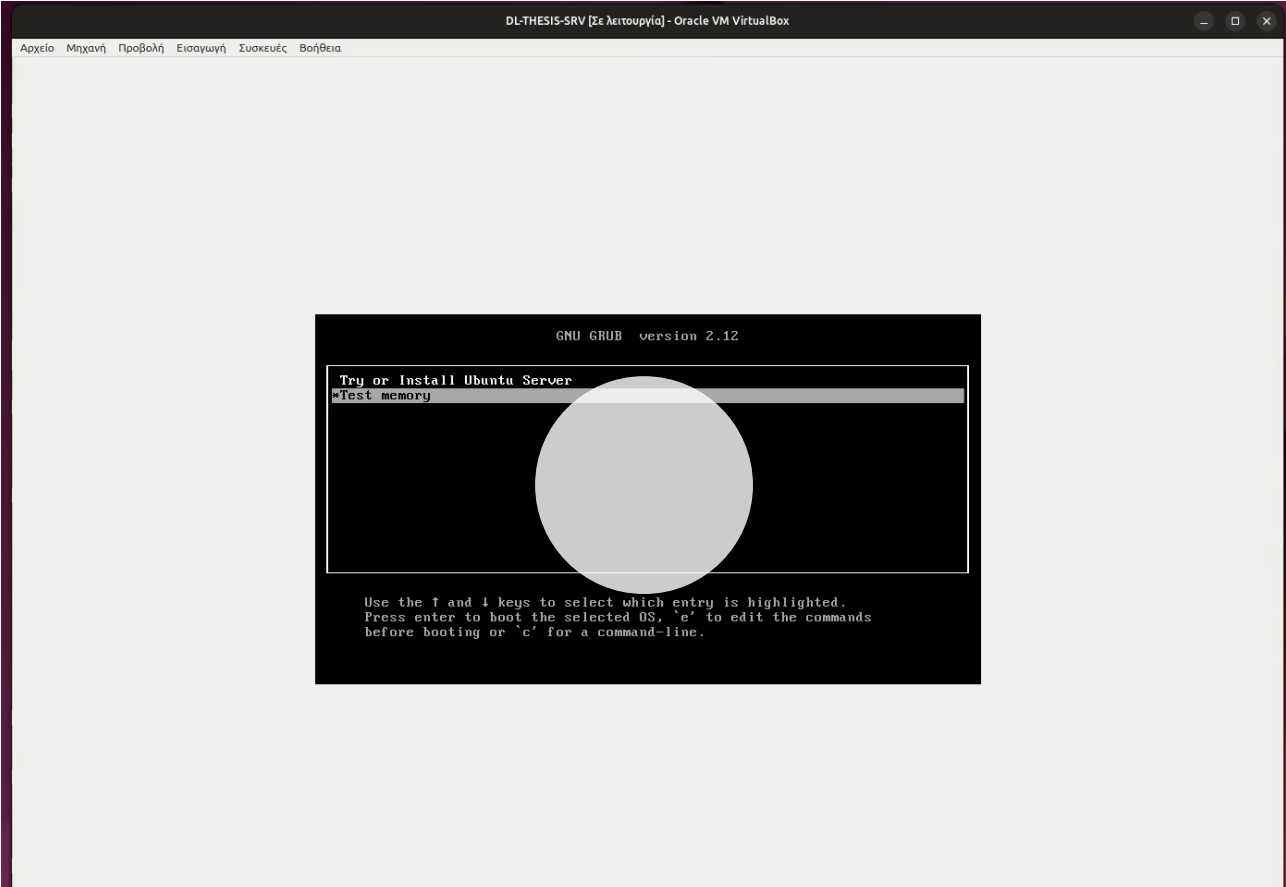
Firewall

Το firewall (ufw) είναι ενεργοποιημένο με αποδοχή των εξωτερικών συνδέσεων μέσω του πρωτοκόλλου SSH.

To	Action	From
22/tcp	ALLOW IN	Anywhere

Βιντεοδηγός

DL-THESIS-SRV (Εγκατάσταση)



0:00 / 14:07

Lxd-Lxc commands

👤 Ευαγγελία Κατσιαντώνη 📅 12/15/24 🌐 Linux 🛠️ commands, cli, lxd, lxc ⌚ Περίπου 41 λεπτά

Εισαγωγή

Το LXD, σε συνδυασμό με το εργαλείο γραμμής εντολών του, το lxc, αποτελεί ένα προηγμένο σύστημα διαχείρισης κοντέινερ, σχεδιασμένο να παρέχει ευελιξία και αποδοτικότητα στη διαχείριση εφαρμογών και υπηρεσιών. Σε αυτό το άρθρο, παρουσιάζονται συνοπτικά οι βασικές εντολές για τη δημιουργία, χρήση, έλεγχο, διαχείριση και δημιουργία αντιγράφων ασφαλείας κοντέινερ, οργανωμένες σε θεματικές ενότητες.

Η ανάλυση αυτή επικεντρώνεται στη λειτουργική αξιοποίηση του LXD, προσφέροντας πρακτική καθοδήγηση για χρήστες με γνώσεις πληροφορικής. Η εμβάθυνση στις τεχνολογίες εικονικοποίησης και κοντεϊνερποίησης, που εξετάζεται εκτενώς παρακάτω, παρέχει το πλαίσιο για την αποτελεσματική χρήση του εργαλείου αυτού.

Note

Στην παρουσίαση των παρακάτω εντολών της γραμμή τερματικού, λαμβάνουμε υπόψη ότι το LXD είναι εγκατεστημένο μέσω Snap σε Sever με λειτουργικό σύστημα Ubuntu 24.04 LTS.

Πλήρης Πίνακας Εντολών Χρήσης - Διαχείρισης

Ενέργεια	Εντολή
Διαχείριση LXD	
Εγκατάσταση LXD	<code>sudo apt install lxd</code>
Εκκίνηση του LXD	<code>lxd init</code>
Επαλήθευση κατάστασης του LXD	<code>lxc-checkconfig</code>
Προβολή διαθέσιμων αποθηκευτικών χώρων	<code>lxc storage list</code>

Ενέργεια	Εντολή
Προσθήκη αποθηκευτικού χώρου	<code>lxc storage create <name> dir</code>
Δημιουργία προφίλ δικτύου	<code>lxc network create <network-name></code>
Προβολή προφίλ	<code>lxc profile show <profile-name></code>
Προβολή λίστας προφίλ	<code>lxc profile list</code>
Προσαρμογή προφίλ	<code>lxc profile edit <profile-name></code>
Δημιουργία project	<code>lxc project create <project-name></code>
Διαγραφή project	<code>lxc project delete <project-name></code>
Δημιουργία image από container	<code>lxc publish <container-name> --alias <image-alias></code>
Διαχείριση Containers	
Δημιουργία container	<code>lxc launch <image> <container-name> -c description="<comment>"</code>
Προβολή διαθέσιμων images	<code>lxc image list</code>
Προβολή ενεργών containers	<code>lxc list</code>
Σύνδεση σε container	<code>lxc exec <container-name> -- /bin/bash</code>
Διακοπή container	<code>lxc stop <container-name></code>
Έναρξη container	<code>lxc start <container-name></code>
Κατάργηση container	<code>lxc delete <container-name></code>
Δημιουργία στιγμιότυπου (snapshot)	<code>lxc snapshot <container-name> <snapshot-name> -c description="<comment>"</code>
Επαναφορά από στιγμιότυπο	<code>lxc restore <container-name> <snapshot-name></code>
Εξαγωγή container	<code>lxc export <container-name> <file-name.tar.gz></code>

Ενέργεια	Εντολή
Εισαγωγή container	<code>lxc import <file-name.tar.gz></code>
Μετονομασία container	<code>lxc rename <old-name> <new-name></code> ή <code>lxc move <old-name> <new-name></code>
Δημιουργία αντιγράφου container	<code>lxc copy <source-container> <new-container-name></code>
Προβολή πόρων container	<code>lxc info <container-name></code>
Διαχείριση Αρχείων	
Αντιγραφή αρχείου στο container	<code>lxc file push <local-path> <container-name>/<path-in-container></code>
Αντιγραφή αρχείου από container	<code>lxc file pull <container-name>/<path-in-container> <local-path></code>
Διαγραφή αρχείου στο container	<code>lxc file delete <container-name>/<path-in-container></code>
Προβολή αρχείων στο container	<code>lxc file list <container-name>/<path-in-container></code>
Επεξεργασία αρχείου στο container	<code>lxc file edit <container-name>/<path-in-container></code>
Διαχείριση Δικτύων	
Δημιουργία προφίλ δικτύου	<code>lxc network create <network-name></code>
Προβολή ενεργών δικτύων	<code>lxc network list</code>
Επεξεργασία δικτύου	<code>lxc network edit <network-name></code>
Διαγραφή δικτύου	<code>lxc network delete <network-name></code>
Προσθήκη δικτύου σε container	<code>lxc network attach <network-name> <container-name></code>
Ασφάλεια και Χρήστες	
Προσθήκη χρήστη	<code>lxc config trust add <file.crt></code>

Ενέργεια	Εντολή
Κατάργηση χρήστη	<code>lxc config trust remove <fingerprint></code>
Προβολή πιστοποιητικών χρηστών	<code>lxc config trust list</code>
Διαχείριση Πόρων	
Περιορισμός μνήμης σε container	<code>lxc config set <container-name> limits.memory <value></code>
Περιορισμός CPU σε container	<code>lxc config set <container-name> limits.cpu <value></code>
Περιορισμός αποθηκευτικού χώρου	<code>lxc config device set <container-name> root size <value></code>
Προσθήκη επιπλέον αποθηκευτικού χώρου	<code>lxc config device add <container-name> <disk-name> disk source=<path></code>
Διαχείριση δικαιωμάτων πόρων	<code>lxc config set <container-name> security.privileged true/false</code>
Ενημερώσεις και Συντήρηση	
Ενημέρωση LXD	<code>sudo snap refresh lxd</code>
Αναβάθμιση container	<code>lxc exec <container-name> -- apt update && apt upgrade</code>
Καθαρισμός μη χρησιμοποιούμενων images	<code>lxc image delete <image-name></code>
Καθαρισμός στιγμιότυπων	<code>lxc delete <container-name>/<snapshot-name></code>
Διαχείριση Nodes (Cluster)	
Δημιουργία cluster	<code>lxd init --cluster</code>
Προσθήκη node στο cluster	<code>lxc cluster add <node-name></code>
Αφαίρεση node από cluster	<code>lxc cluster remove <node-name></code>
Προβολή κατάστασης cluster	<code>lxc cluster list</code>

Διαχείριση LXD

Εκκίνηση του LXD

Η εντολή `lxd init` είναι η αρχική εντολή για τη ρύθμιση του LXD στο σύστημά σας. Εκτελείται μόνο μία φορά κατά την εγκατάσταση και ρυθμίζει το περιβάλλον του LXD.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εκτελεί τη βασική ρύθμιση του LXD, δημιουργώντας το κατάλληλο περιβάλλον για containers.
- Ρυθμίζει τις παραμέτρους του LXD, όπως αποθηκευτικό χώρο, δίκτυο και άλλα.

2. Ορίσματα και Ρυθμίσεις:

- Κατά την εκτέλεση της εντολής, ο χρήστης θα καθορίσει διάφορες ρυθμίσεις μέσω παραμέτρων:
 - **Storage backend:** Επιλογή του τύπου αποθηκευτικού χώρου (π.χ., `dir`, `zfs`, `lvm`).
 - **Networking:** Ρύθμιση δικτύου, π.χ., δημιουργία μιας προεπιλεγμένης εικονικής διασύνδεσης.
 - **Cluster settings:** Ρύθμιση για χρήση του LXD σε περιβάλλον cluster (αν χρειάζεται).
 - **Security options:** Ρύθμιση για ασφάλεια μέσω AppArmor ή SELinux.
 - **Password:** Ρύθμιση του κωδικού για τον root χρήστη του LXD.
 - **Storage location:** Ορισμός της τοποθεσίας για την αποθήκευση των containers.

3. Πληροφορίες που Παρέχονται:

- Εμφανίζει οδηγίες βημάτων για τη διαμόρφωση του LXD, και ρωτά τον χρήστη για τις παραμέτρους που θα χρησιμοποιηθούν.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν γίνεται η πρώτη εγκατάσταση του LXD ή όταν απαιτείται επαναρύθμιση.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxd init
Would you like to configure a new storage pool (yes/no)? yes
Name of the storage pool: default
Create a new network bridge (yes/no)? yes
Do you want to configure the LXD daemon to start automatically (yes/no)? yes
```

Note

Όλες οι ρυθμίσεις μπορούν να αλλάξουν αργότερα μέσω του `lxd init` ή άλλων εργαλείων διαχείρισης.

Επαλήθευση κατάστασης του LXD

Η εντολή `lxc-checkconfig` ελέγχει αν το σύστημα πληροί τις απαιτήσεις για τη σωστή λειτουργία του LXD, επαληθεύοντας τις ρυθμίσεις του kernel και άλλες σημαντικές παραμέτρους.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Ελέγχει αν το σύστημα διαθέτει τις απαιτούμενες δυνατότητες για την εκτέλεση του LXD, όπως υποστήριξη για namespaces, cgroups, AppArmor και άλλες ρυθμίσεις του kernel.
- Η εντολή αναφέρει αν κάποια εξαρτήματα λείπουν ή αν δεν είναι σωστά ρυθμισμένα, παρέχοντας προτάσεις για την επίλυση των προβλημάτων.

2. Ορίσματα και Ρυθμίσεις:

- Δεν υπάρχουν ειδικά ορίσματα ή ρυθμίσεις, η εντολή εκτελείται απλώς χωρίς παραμέτρους και επιστρέφει τις απαραίτητες πληροφορίες.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφει πληροφορίες για τα χαρακτηριστικά του kernel που είναι ενεργοποιημένα ή απενεργοποιημένα, όπως:
 - **Namespaces:** Αν υποστηρίζονται.
 - **Cgroups:** Αν υποστηρίζονται και ποια έκδοση (π.χ., v2).
 - **AppArmor:** Αν είναι ενεργοποιημένο και αν είναι ρυθμισμένο σωστά.
 - **Kernel configuration:** Αν δεν βρεθεί το αρχείο διαμόρφωσης του kernel, η εντολή

θα προσπαθήσει να το βρει από άλλες τοποθεσίες.

- Σε περίπτωση προβλημάτων, το LXD θα παρέχει προτάσεις για να τα διορθώσεις.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να ελέγξεις αν το σύστημα πληροί τις απαιτήσεις του LXD πριν την εγκατάσταση ή τη χρήση του. Είναι ιδιαίτερα χρήσιμο σε νέες εγκαταστάσεις ή σε περιβάλλοντα παραγωγής για να διασφαλίσεις ότι το σύστημα είναι σωστά ρυθμισμένο.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc-checkconfig
Kernel configuration not found at /proc/config.gz; searching other
paths...
--- Namespaces ---
Namespaces: enabled
Cgroup v2: enabled
AppArmor: enabled
```

6. Επεξήγηση της Εξόδου:

- **Namespaces: enabled** : Τα namespaces του kernel είναι ενεργοποιημένα, κάτι που είναι απαραίτητο για τη σωστή απομόνωση των containers.
- **Cgroup v2: enabled** : Υποστηρίζεται η έκδοση 2 των cgroups, η οποία είναι απαραίτητη για τον περιορισμό πόρων των containers.
- **AppArmor: enabled** : Το AppArmor είναι ενεργοποιημένο για την ασφάλεια των containers.
- **Kernel configuration not found** : Αν δεν βρεθεί το αρχείο διαμόρφωσης του kernel, μπορεί να χρειαστεί να το ενεργοποιήσεις χειροκίνητα ή να το εγκαταστήσεις μέσω του πυρήνα.

Note

Αν κάποιο από τα απαιτούμενα χαρακτηριστικά του kernel είναι απενεργοποιημένο, μπορείς να το ενεργοποιήσεις είτε μέσω του πυρήνα (π.χ., αναβάθμιση kernel), είτε με την προσαρμογή των ρυθμίσεων στο σύστημά σου.

Αν χρειάζεστε την πλήρη επικόλληση ή τροποποιήσεις στο παραπάνω, ενημερώστε με!

Προβολή διαθέσιμων αποθηκευτικών χώρων

Η εντολή `lxc storage list` προβάλλει τα διαθέσιμα αποθηκευτικά μέσα για το LXD, παρέχοντας πληροφορίες για την τρέχουσα κατάσταση και τις δυνατότητες κάθε αποθηκευτικού χώρου.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εμφανίζει τα διαθέσιμα αποθηκευτικά pools και τους τύπους τους. Αυτά τα αποθηκευτικά μέσα χρησιμοποιούνται για την αποθήκευση των containers και άλλων δεδομένων του LXD.
- Παρέχει πληροφορίες για το αν είναι ενεργοποιημένα, τη χρήση τους και τη συνολική χωρητικότητα.

2. Ορίσματα και Ρυθμίσεις:

- Η εντολή δεν απαιτεί παραμέτρους και εκτελείται απευθείας για να εμφανίσει την κατάσταση των αποθηκευτικών χώρων. Ωστόσο, μπορείς να χρησιμοποιήσεις παραμέτρους για πιο συγκεκριμένες πληροφορίες:
 - `-c` : Καθορίζει ποιο περιβάλλον (context) να χρησιμοποιηθεί για την εμφάνιση των πληροφοριών.
 - `--format` : Καθορίζει τη μορφή εξόδου (π.χ., JSON ή YAML).

3. Πληροφορίες που Παρέχονται:

- Εμφανίζει μια λίστα με τα διαθέσιμα αποθηκευτικά pools, μαζί με τα εξής στοιχεία:
 - **NAME**: Όνομα του αποθηκευτικού pool.
 - **DRIVER**: Τύπος του αποθηκευτικού μέσου (π.χ., `dir` , `lvm` , `zfs`).
 - **SOURCE**: Η τοποθεσία αποθήκευσης (π.χ., κατάλογος ή φυσικός δίσκος).
 - **STATE**: Κατάσταση του pool (π.χ., `CREATED` , `MOUNTED`).
 - **USED BY**: Ο αριθμός των containers ή άλλων πόρων που χρησιμοποιούν το pool.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δεις τα διαθέσιμα αποθηκευτικά μέσα και να επιβεβαιώσεις την κατάσταση και τη χρήση τους πριν από την εκκίνηση ή τη διαχείριση containers.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc storage list
+-----+-----+-----+-----+
| NAME      | DRIVER | SOURCE  | STATE   | USED BY |
+-----+-----+-----+-----+-----+
```

default	btrfs	/dev/sda	CREATED	1	
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

Note

Αν δεν υπάρχει αποθηκευτικός χώρος, μπορείς να δημιουργήσεις έναν νέο χρησιμοποιώντας την εντολή `lxc storage create <name> <driver> .`

Προσθήκη αποθηκευτικού χώρου

Η εντολή `lxc storage create` δημιουργεί ένα νέο αποθηκευτικό pool για το LXD.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί ένα νέο αποθηκευτικό pool για χρήση με το LXD, επιτρέποντας την αποθήκευση containers και άλλων πόρων.

2. Ορίσματα και Ρυθμίσεις:

- **<name>** : Όνομα του αποθηκευτικού pool.
- **<driver>** : Ο τύπος του αποθηκευτικού μέσου (π.χ., `dir` , `lvm` , `zfs`).
- Ενδεικτικά παραδείγματα τύπων:
 - `dir` : Χρησιμοποιεί έναν κανονικό κατάλογο στο σύστημα για αποθήκευση.
 - `lvm` : Χρησιμοποιεί LVM για αποθηκευτικούς χώρους.
 - `zfs` : Χρησιμοποιεί το σύστημα αρχείων ZFS για αποθηκευτικούς χώρους.

3. Πληροφορίες που Παρέχονται:

- Δημιουργεί ένα νέο αποθηκευτικό pool σύμφωνα με τις ρυθμίσεις που παρέχεις. Αν το pool δημιουργηθεί με επιτυχία, θα επιστραφεί μήνυμα επιβεβαίωσης.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δημιουργήσεις έναν νέο αποθηκευτικό χώρο για το LXD, για να αποθηκεύσεις containers ή άλλα δεδομένα.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc storage create mypool dir
```

Note

Αν το storage pool με το ίδιο όνομα υπάρχει ήδη, θα λάβεις μήνυμα σφάλματος.

Δημιουργία προφίλ δικτύου

Η εντολή `lxc network create <network-name>` δημιουργεί ένα νέο προφίλ δικτύου για χρήση με containers στο LXD. Επιτρέπει τη διαχείριση των δικτυακών παραμέτρων, όπως η δημιουργία bridges, το VLAN tagging και η ρύθμιση της δικτυακής υποδομής για containers.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί ένα νέο προφίλ δικτύου που μπορεί να χρησιμοποιηθεί από containers στο LXD.
- Το δίκτυο μπορεί να είναι τύπου `bridge`, `macvlan`, `ipvlan`, ή `sriov`, και καθορίζει τον τρόπο σύνδεσης των containers με το φυσικό ή εικονικό δίκτυο.

2. Ορίσματα και Ρυθμίσεις:

- **<network-name>** : Το όνομα του δικτύου που θέλεις να δημιουργήσεις.
- Υποστηρίζει και άλλες παραμέτρους για να προσδιορίσεις τις λεπτομέρειες του δικτύου. Για παράδειγμα, η επιλογή `type=bridge` για την αναφορά σε ένα εικονικό bridge.

Ενδεικτικά, μπορείς να καθορίσεις ρυθμίσεις όπως:

- **type** : Ο τύπος του δικτύου, π.χ., `bridge`, `macvlan`, `ipvlan`.
- **bridge.mode** : Ο τρόπος λειτουργίας του bridge, π.χ., `NAT`, `fan`, κ.λπ.
- **ipv4.address** : Η διεύθυνση IP για το δίκτυο.
- **ipv6.address** : Η διεύθυνση IPv6 για το δίκτυο.

3. Πληροφορίες που Παρέχονται:

- Δημιουργείται το προφίλ δικτύου και επιστρέφεται μήνυμα επιτυχίας ή σφάλματος.
- Το νέο δίκτυο μπορεί να χρησιμοποιηθεί για να συνδέσεις containers.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δημιουργήσεις ένα νέο δίκτυο για containers, είτε

για χρήση σε εσωτερικά δίκτυα (π.χ., για σύνδεση containers μεταξύ τους) είτε για σύνδεση στο εξωτερικό δίκτυο μέσω bridge.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc network create mybridge type=bridge  
bridge.mode=fan
```

Note

- Η δημιουργία ενός δικτύου τύπου `bridge` επιτρέπει στους containers να χρησιμοποιούν NAT για πρόσβαση στο εξωτερικό δίκτυο. Μπορείς επίσης να χρησιμοποιήσεις άλλους τύπους δικτύου, ανάλογα με τις ανάγκες σου (π.χ., `macvlan` για απομόνωση ή `ipvlan` για καλύτερη απόδοση).
- Εάν χρησιμοποιείς `macvlan` ή `ipvlan`, το LXD ενδέχεται να απαιτεί ειδικές ρυθμίσεις στο φυσικό δίκτυο.

Προβολή προφίλ

Η εντολή `lxc profile show <profile-name>` εμφανίζει τις ρυθμίσεις του επιλεγμένου προφίλ για το LXD. Το προφίλ ορίζει τις παραμέτρους που ισχύουν για τα containers που το χρησιμοποιούν, όπως δικτυακές ρυθμίσεις, περιορισμοί πόρων, και άλλες παραμέτρους.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εμφανίζει τις ρυθμίσεις ενός συγκεκριμένου προφίλ του LXD.
- Παρουσιάζει λεπτομέρειες για τις προσαρμοσμένες ρυθμίσεις δικτύου, περιορισμούς πόρων, καθώς και άλλες παραμέτρους που ισχύουν για τα containers που χρησιμοποιούν το εν λόγω προφίλ.

2. Ορίσματα και Ρυθμίσεις:

- `<profile-name>` : Όνομα του προφίλ που θέλεις να δεις.
- Εάν δεν παρέχεται όνομα προφίλ, η εντολή δεν θα λειτουργήσει και θα επιστρέψει σφάλμα.

3. Πληροφορίες που Παρέχονται:

- Περιλαμβάνει πληροφορίες για:
 - **Networking:** Ρυθμίσεις δικτύου που ισχύουν για το προφίλ.
 - **Limits:** Ορισμοί περιορισμών πόρων, όπως CPU, μνήμη, κλπ.
 - **Devices:** Λίστα με τα συσκευές που είναι διαθέσιμες για χρήση από το προφίλ.
 - **Security:** Ρυθμίσεις ασφαλείας που σχετίζονται με το προφίλ.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να δεις ή να ελέγξεις τις ρυθμίσεις του προφίλ πριν τη δημιουργία ενός container που θα χρησιμοποιήσει το εν λόγω προφίλ.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc profile show default
config:
  limits.memory: 2GB
  limits.cpu: "2"
  security.privileged: "false"
devices:
  eth0:
    name: eth0
    nictype: bridged
    parent: lxdbr0
    type: nic
```

Note

Αν δεν έχει οριστεί προφίλ, το LXD χρησιμοποιεί το προφίλ `default` από προεπιλογή.

Προσαρμογή προφίλ

Η εντολή `lxc profile edit <profile-name>` επιτρέπει την επεξεργασία των ρυθμίσεων ενός υπάρχοντος προφίλ στο LXD. Αυτή η εντολή ανοίγει τον επεξεργαστή κειμένου για να τροποποιήσεις τις παραμέτρους του προφίλ.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Επιτρέπει την επεξεργασία ενός προφίλ μέσω ενός επεξεργαστή κειμένου (π.χ., `vim`, `nano`).

- Τροποποιούνται ρυθμίσεις για το δίκτυο, περιορισμούς πόρων, συσκευές και άλλες παραμέτρους που ισχύουν για τα containers που χρησιμοποιούν το προφίλ.

2. Ορίσματα και Ρυθμίσεις:

- **<profile-name>** : Το όνομα του προφίλ που θέλεις να επεξεργαστείς.
- Η εντολή ανοίγει τον επεξεργαστή κειμένου για να επεξεργαστείς τις ρυθμίσεις του προφίλ.

3. Πληροφορίες που Παρέχονται:

- Το περιεχόμενο του προφίλ εμφανίζεται στον επεξεργαστή κειμένου, και μπορείς να το τροποποιήσεις σε πραγματικό χρόνο.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν χρειάζεσαι να τροποποιήσεις ένα προφίλ για να προσαρμόσεις τις ρυθμίσεις του LXD.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc profile edit default
```

Note

Οι αλλαγές που γίνονται με αυτήν την εντολή ισχύουν αμέσως και επηρεάζουν τα containers που χρησιμοποιούν το προφίλ.

Δημιουργία project

Η εντολή `lxc project create <project-name>` δημιουργεί ένα νέο project στο LXD. Τα projects χρησιμοποιούνται για την οργάνωση και διαχείριση των containers και των πόρων του LXD.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί ένα νέο project για τη διαχείριση containers και αποθηκευτικών πόρων μέσα στο LXD.
- Τα projects χρησιμοποιούνται για την οργάνωση των πόρων σε ξεχωριστές ενότητες.

2. Ορίσματα και Ρυθμίσεις:

- **<project-name>** : Το όνομα του νέου project που θα δημιουργηθεί.
- Η εντολή δημιουργεί ένα project με το καθορισμένο όνομα και καταχωρεί τις ρυθμίσεις για αυτό.

3. Πληροφορίες που Παρέχονται:

- Δημιουργείται ένα νέο project στο σύστημα και επιστρέφεται μήνυμα επιβεβαίωσης.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να οργανώσεις τα containers και τους πόρους του LXD σε ξεχωριστά projects.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc project create myproject
```

Note

Μετά τη δημιουργία του project, μπορείς να αναθέσεις containers και άλλους πόρους σε αυτό το project.

Διαγραφή project

Η εντολή `lxc project delete <project-name>` διαγράφει το καθορισμένο project από το LXD, μαζί με όλα τα δεδομένα που ανήκουν σε αυτό.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Διαγράφει το project και όλους τους πόρους που συνδέονται με αυτό.
- Πρέπει να είσαι προσεκτικός καθώς η διαγραφή του project μπορεί να οδηγήσει σε απώλεια δεδομένων.

2. Ορίσματα και Ρυθμίσεις:

- **<project-name>** : Το όνομα του project που θέλεις να διαγράψεις.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφεται μήνυμα επιβεβαίωσης διαγραφής του project και των πόρων του.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να διαγράψεις ένα project από το LXD.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc project delete myproject
```

Note

Η διαγραφή ενός project είναι μη αναστρέψιμη και πρέπει να γίνει με προσοχή.

Δημιουργία image από container

Η εντολή `lxc publish <container-name> --alias <image-alias>` δημιουργεί μια εικόνα (image) από έναν container και τη δημοσιεύει με ένα alias.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί μια εικόνα από έναν container, επιτρέποντας την αναπαραγωγή του container σε άλλους hosts.
- Το alias χρησιμοποιείται για να αναφερθείς στην εικόνα αντί για το μοναδικό ID της.

2. Ορίσματα και Ρυθμίσεις:

- **<container-name>** : Το όνομα του container από τον οποίο θα δημιουργηθεί η εικόνα.
- **--alias <image-alias>** : Το alias για την εικόνα (π.χ., `ubuntu-20.04`).

3. Πληροφορίες που Παρέχονται:

- Δημιουργείται η εικόνα του container και επιστρέφεται το ID της εικόνας και το alias που καθορίστηκε.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δημιουργήσεις μια εικόνα από έναν container και να τη δημοσιεύσεις για επαναχρησιμοποίηση ή να τη μοιραστείς με άλλους χρήστες.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc publish mycontainer --alias ubuntu-20.04
```

Note

Μπορείς να χρησιμοποιήσεις την εικόνα που δημιουργήθηκε με την εντολή `lxc image list` για να τη χρησιμοποιήσεις σε νέους containers.

Διαχείριση Containers

Δημιουργία container

Η εντολή `lxc launch <image> <container-name> -c description="<comment>"` δημιουργεί και εκκινεί ένα νέο container από την επιλεγμένη εικόνα (image).

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί ένα νέο container και το ξεκινάει από μια διαθέσιμη εικόνα.

2. Ορίσματα και Ρυθμίσεις:

- **<image>** : Η εικόνα από την οποία θα δημιουργηθεί το container (π.χ., `ubuntu:20.04`).
- **<container-name>** : Το όνομα που θα δοθεί στο νέο container.
- **-c description="<comment>"** : Προαιρετικό πεδίο για την προσθήκη μιας περιγραφής του container.

3. Πληροφορίες που Παρέχονται:

- Δημιουργία του container και επιστροφή του ονόματος του νέου container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για την εκκίνηση ενός νέου container με βάση μια εικόνα.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc launch ubuntu:20.04 mycontainer -c description="My first container"
```

Note

Η εικόνα μπορεί να προέρχεται από την τοπική αποθήκη εικόνας ή από μια απομακρυσμένη πηγή.

Προβολή διαθέσιμων images

Η εντολή `lxc image list` εμφανίζει τη λίστα των διαθέσιμων εικόνων (images) που μπορείς να χρησιμοποιήσεις για τη δημιουργία containers.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Παρουσιάζει τη λίστα των εικόνων που είναι διαθέσιμες για χρήση στην πλατφόρμα LXD.

2. Πληροφορίες που Παρέχονται:

- Εμφανίζεται λίστα με εικόνες που περιλαμβάνει το όνομα της εικόνας, την έκδοση, τον τύπο και την κατάσταση.

3. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δεις ποιες εικόνες είναι διαθέσιμες για χρήση.

4. Παράδειγμα Χρήσης:

```
root@server:~# lxc image list
```

Note

Μπορείς να δεις τις εικόνες που είναι αποθηκευμένες τοπικά ή να κατεβάσεις νέες εικόνες από το remote LXD server.

Προβολή ενεργών containers

Η εντολή `lxc list` εμφανίζει τα containers που είναι ενεργά αυτή τη στιγμή στο σύστημα.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εμφανίζει τη λίστα των containers που είναι ενεργά και τρέχουν στο σύστημα.

2. Πληροφορίες που Παρέχονται:

- Εμφανίζεται κατάσταση για κάθε container, όπως η κατάσταση εκκίνησης, οι διευθύνσεις IP και οι πόροι.

3. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να δεις ποια containers είναι ενεργά και ποια είναι η κατάσταση τους.

4. Παράδειγμα Χρήσης:

```
root@server:~# lxc list
```

Note

Μπορείς να προσθέσεις φίλτρα για να περιορίσεις τη λίστα, π.χ., για να εμφανίσεις μόνο τα ενεργά ή τα σταματημένα containers.

Σύνδεση σε container

Η εντολή `lxc exec <container-name> -- /bin/bash` επιτρέπει στον χρήστη να συνδεθεί σε ένα container και να εκτελέσει εντολές μέσα σε αυτό.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Συνδέεται σε ένα container και ανοίγει έναν τερματικό shell για να εκτελέσεις εντολές στο περιβάλλον του container.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Το όνομα του container στο οποίο θέλεις να συνδεθείς.
- `-- /bin/bash` : Εκτελεί το Bash shell μέσα στο container.

3. Πληροφορίες που Παρέχονται:

- Ανοίγει το Bash shell και επιτρέπει στον χρήστη να εκτελεί εντολές στο container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να έχεις πρόσβαση στο περιβάλλον του container για διαχείριση ή εκτέλεση εντολών.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc exec mycontainer -- /bin/bash
```

Note

Αν δεν υπάρχει εγκατεστημένο το Bash, μπορείς να χρησιμοποιήσεις άλλους διαχειριστές shell, π.χ., `sh` .

Διακοπή container

Η εντολή `lxc stop <container-name>` σταματάει την εκτέλεση ενός container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Σταματά το container χωρίς να το διαγράψει ή να το καταργήσει.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Το όνομα του container που θέλεις να σταματήσεις.

3. Πληροφορίες που Παρέχονται:

- Αποκρίνεται με την κατάσταση του container μετά την προσπάθεια διακοπής.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να σταματήσεις την εκτέλεση ενός container, αλλά να το διατηρήσεις για μελλοντική εκκίνηση.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc stop mycontainer
```

Note

Η εντολή μπορεί να χρειαστεί να αναγνωρίσει παραμέτρους για τη διακοπή της διαδικασίας με επιτυχία.

Έναρξη container

Η εντολή `lxc start <container-name>` εκκινεί ένα container που έχει σταματήσει.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εκκινεί το container που έχει σταματήσει και ξεκινά την εκτέλεσή του.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Το όνομα του container που θέλεις να ξεκινήσεις.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφει την κατάσταση του container μετά την εκκίνηση.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να ξεκινήσεις ένα container που έχει σταματήσει και θέλεις να το επανεκκινήσεις.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc start mycontainer
```

Note

Εάν το container δεν έχει σταματήσει σωστά ή υπάρχει πρόβλημα με την εκκίνηση, η εντολή ενδέχεται να αποτύχει και να επιστρέψει σχετικό μήνυμα σφάλματος.

Κατάργηση container

Η εντολή `lxc delete <container-name>` καταργεί ένα container και όλα τα δεδομένα του.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Καταργεί μόνιμα το container και όλα τα δεδομένα του, χωρίς να επηρεάζει τα αποθηκευμένα images.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Το όνομα του container που θέλεις να καταργήσεις.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφεται μήνυμα επιτυχίας ή σφάλματος που αναφέρεται στην κατάργηση του container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να διαγράψεις οριστικά ένα container που δεν είναι πλέον χρήσιμο.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc delete mycontainer
```

Note

Η εντολή καταργεί το container μόνιμα. Αν θες να διατηρήσεις το container, χρησιμοποίησε την εντολή `lxc stop` για να το σταματήσεις χωρίς να το διαγράψεις.

Δημιουργία στιγμιότυπου (snapshot)

Η εντολή `lxc snapshot <container-name> <snapshot-name> -c description="<comment>"` δημιουργεί ένα στιγμιότυπο (snapshot) του container σε μια συγκεκριμένη χρονική στιγμή.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί ένα στιγμιότυπο του container που περιλαμβάνει την κατάσταση του συστήματος, τα δεδομένα και τις ρυθμίσεις του container.

2. Ορίσματα και Ρυθμίσεις:

- **<container-name>** : Το όνομα του container για το οποίο θέλεις να δημιουργήσεις το στιγμιότυπο.
- **<snapshot-name>** : Το όνομα του στιγμιότυπου που θα δημιουργηθεί.
- **-c description="<comment>"** : Προαιρετικό πεδίο για την προσθήκη μιας περιγραφής στο στιγμιότυπο.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφεται μήνυμα επιτυχίας με το όνομα του νέου στιγμιότυπου και τις λεπτομέρειες του.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να αποθηκεύσεις την τρέχουσα κατάσταση του container για μελλοντική αποκατάσταση.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc snapshot mycontainer snapshot1 -c  
description="Initial snapshot"
```

Note

Τα στιγμιότυπα είναι χρήσιμα για την αποκατάσταση του container σε προηγούμενη κατάσταση σε περίπτωση προβλημάτων.

Επαναφορά από στιγμιότυπο

Η εντολή `lxc restore <container-name> <snapshot-name>` επαναφέρει το container στην κατάσταση του συγκεκριμένου στιγμιότυπου.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Επαναφέρει το container στην κατάσταση που είχε τη στιγμή που δημιουργήθηκε το στιγμιότυπο.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Το όνομα του container που θέλεις να επαναφέρεις.
- `<snapshot-name>` : Το όνομα του στιγμιότυπου από το οποίο θα γίνει η επαναφορά.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφεται μήνυμα επιτυχίας αν η επαναφορά ήταν επιτυχής.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θες να επιστρέψεις σε μια προηγούμενη κατάσταση του container μετά από αλλαγές ή σφάλματα.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc restore mycontainer snapshot1
```

Note

Η επαναφορά καταργεί τις αλλαγές που έγιναν μετά το δημιουργία του στιγμιότυπου.

Εξαγωγή container

Η εντολή `lxc export <container-name> <file-name.tar.gz>` εξαγει το container σε ένα αρχείο αποθήκευσης.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί ένα αρχείο TAR που περιέχει το σύνολο των δεδομένων του container και των ρυθμίσεών του.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Το όνομα του container που θέλεις να εξάγεις.
- `<file-name.tar.gz>` : Το όνομα και η διαδρομή του αρχείου που θα αποθηκευτεί.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφεται το όνομα του εξαγόμενου αρχείου και η διαδρομή του στον δίσκο.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δημιουργήσεις αντίγραφο ασφαλείας ενός container ή να το μεταφέρεις σε άλλο σύστημα.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc export mycontainer mycontainer-backup.tar.gz
```

Note

Το εξαγόμενο αρχείο μπορεί να εισαχθεί ξανά σε άλλο σύστημα μέσω της εντολής `lxc import` .

Εισαγωγή container

Η εντολή `lxc import <file-name.tar.gz>` εισάγει ένα container από ένα αρχείο αποθήκευσης (π.χ., TAR αρχείο).

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εισάγει το container από ένα αρχείο αποθήκευσης (π.χ., αρχείο TAR) και το επαναφέρει στο σύστημα.

2. Ορίσματα και Ρυθμίσεις:

- `<file-name.tar.gz>` : Το αρχείο από το οποίο θα γίνει η εισαγωγή του container.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφεται μήνυμα επιτυχίας με το όνομα του εισαγόμενου container και τη νέα του κατάσταση.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να εισάγεις ένα container που έχει εξαχθεί από άλλο σύστημα ή έχει δημιουργηθεί ως αντίγραφο ασφαλείας.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc import mycontainer-backup.tar.gz
```

Note

Αν το αρχείο TAR περιέχει πολλά containers ή άλλες ρυθμίσεις, αυτές θα επαναφερθούν κατά την εισαγωγή.

Μετονομασία container

Η εντολή `lxc rename <old-name> <new-name>` ή `lxc move <old-name> <new-name>` μετονομάζει ή μετακινεί ένα container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Μετονομάζει το container αλλάζοντας το όνομά του.
- Η εντολή `move` μπορεί να χρησιμοποιηθεί και για να μετακινήσεις το container σε διαφορετική τοποθεσία (π.χ., από έναν χώρο αποθήκευσης σε άλλο).

2. Ορίσματα και Ρυθμίσεις:

- **<old-name>** : Το τρέχον όνομα του container που θες να μετονομάσεις.
- **<new-name>** : Το νέο όνομα για το container.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφεται μήνυμα επιτυχίας με το νέο όνομα του container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δώσεις νέο όνομα σε ένα container ή να το μετακινήσεις σε άλλη τοποθεσία.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc rename oldcontainer newcontainer
```

Note

Αν χρησιμοποιηθεί η εντολή `move`, το container μπορεί να μετακινηθεί σε διαφορετικό χώρο αποθήκευσης, αλλά το όνομα θα παραμείνει το ίδιο.

Δημιουργία αντιγράφου container

Η εντολή `lxc copy <source-container> <new-container-name>` δημιουργεί ένα αντίγραφο του container με νέο όνομα.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί αντίγραφο ενός υπάρχοντος container, διατηρώντας τις ρυθμίσεις και τα δεδομένα του.

2. Ορίσματα και Ρυθμίσεις:

- **<source-container>** : Το όνομα του container που θέλεις να αντιγράψεις.
- **<new-container-name>** : Το όνομα του νέου container που θα δημιουργηθεί από το αντίγραφο.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφεται μήνυμα επιτυχίας και το νέο όνομα του αντιγράφου του container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δημιουργήσεις ένα αντίγραφο ενός container για δοκιμές ή για backup.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc copy mycontainer mynewcontainer
```

Note

Το αντίγραφο θα έχει τα ίδια δεδομένα και ρυθμίσεις με το αρχικό container, αλλά θα είναι ανεξάρτητο.

Προβολή πόρων container

Η εντολή `lxc info <container-name>` εμφανίζει πληροφορίες σχετικά με την κατάσταση και τους πόρους του container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Παρέχει λεπτομέρειες για την κατάσταση του container, όπως τη χρήση πόρων (CPU, μνήμη), τις ρυθμίσεις δικτύου και την τρέχουσα κατάσταση του container.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Το όνομα του container για το οποίο θες να δεις τις πληροφορίες.

3. Πληροφορίες που Παρέχονται:

- Επιστρέφει μια αναλυτική κατάσταση του container, συμπεριλαμβανομένων των πόρων που χρησιμοποιεί, των δικτύων και άλλων ρυθμίσεων.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να δεις τη χρήση πόρων και τις πληροφορίες κατάστασης του container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc info mycontainer
```

Note

Αν το container δεν εκτελείται, ενδέχεται να μην υπάρχουν πληροφορίες σχετικά με τη χρήση πόρων. Ωστόσο, η κατάσταση του container (εκκίνησης, διακοπής κ.λπ.) θα είναι διαθέσιμη.

Διαχείριση Αρχείων

Αντιγραφή αρχείου στο container

Η εντολή `lxc file push <local-path> <container-name>/<path-in-container>` χρησιμοποιείται για να αντιγράψει ένα αρχείο από το τοπικό σύστημα στο συγκεκριμένο container στον καθορισμένο φάκελο.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Αντιγράφει ένα αρχείο από το τοπικό σύστημα στον φάκελο του container που ορίζεται στην εντολή.

2. Ορίσματα και Ρυθμίσεις:

- `<local-path>` : Διαδρομή του αρχείου στο τοπικό σύστημα.
- `<container-name>` : Όνομα του container στο οποίο θα αντιγραφεί το αρχείο.
- `<path-in-container>` : Διαδρομή στον φάκελο του container όπου θα αποθηκευτεί το αρχείο.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση επιτυχίας ή αποτυχίας της αντιγραφής του αρχείου.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να μεταφέρεις αρχεία από τον τοπικό υπολογιστή στο container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc file push /home/user/file.txt container1/etc/
```

```
file.txt
```

Αντιγραφή αρχείου από container

Η εντολή `lxc file pull <container-name>/<path-in-container> <local-path>` χρησιμοποιείται για να αντιγράψει ένα αρχείο από το container στο τοπικό σύστημα.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Αντιγράφει ένα αρχείο από το container στο τοπικό σύστημα.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Όνομα του container από το οποίο θα αντιγραφεί το αρχείο.
- `<path-in-container>` : Διαδρομή του αρχείου στο container.
- `<local-path>` : Διαδρομή στον τοπικό φάκελο όπου θα αποθηκευτεί το αρχείο.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση επιτυχίας ή αποτυχίας της αντιγραφής του αρχείου.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να μεταφέρεις αρχεία από το container στον τοπικό υπολογιστή.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc file pull container1/etc/file.txt /home/user/  
file.txt
```

Διαγραφή αρχείου στο container

Η εντολή `lxc file delete <container-name>/<path-in-container>` χρησιμοποιείται για να διαγράψει ένα αρχείο μέσα στο container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Διαγράφει το αρχείο από το container στην καθορισμένη διαδρομή.

2. Ορίσματα και Ρυθμίσεις:

- **<container-name>** : Όνομα του container από το οποίο θα διαγραφεί το αρχείο.
- **<path-in-container>** : Διαδρομή του αρχείου μέσα στο container που θέλεις να διαγράψεις.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση της διαγραφής του αρχείου.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να αφαιρέσεις αρχεία από το container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc file delete container1/etc/file.txt
```

Προβολή αρχείων στο container

Η εντολή `lxc file list <container-name>/<path-in-container>` εμφανίζει μια λίστα με τα αρχεία στον καθορισμένο φάκελο του container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εμφανίζει τα αρχεία σε έναν φάκελο του container.

2. Ορίσματα και Ρυθμίσεις:

- **<container-name>** : Όνομα του container.
- **<path-in-container>** : Διαδρομή του φακέλου μέσα στο container που θέλεις να δεις τα περιεχόμενα.

3. Πληροφορίες που Παρέχονται:

- Λίστα των αρχείων στον καθορισμένο φάκελο του container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να δεις τα αρχεία σε έναν φάκελο του container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc file list container1/etc/
```

Επεξεργασία αρχείου στο container

Η εντολή `lxc file edit <container-name>/<path-in-container>` ανοίγει έναν επεξεργαστή για την τροποποίηση του αρχείου στο container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Επιτρέπει την επεξεργασία αρχείων μέσα στο container μέσω ενός επεξεργαστή.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Όνομα του container.
- `<path-in-container>` : Διαδρομή του αρχείου που θέλεις να επεξεργαστείς.

3. Πληροφορίες που Παρέχονται:

- Η εντολή ανοίγει έναν επεξεργαστή για την επεξεργασία του αρχείου στο container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να επεξεργαστείς αρχεία άμεσα στο container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc file edit container1/etc/file.txt
```

Διαχείριση Δικτύων

Δημιουργία προφίλ δικτύου

Η εντολή `lxc network create <network-name>` χρησιμοποιείται για να δημιουργήσει ένα νέο δίκτυο στο LXD.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί ένα νέο δίκτυο στο LXD με το όνομα που καθορίζεται στην εντολή.

2. Ορίσματα και Ρυθμίσεις:

- `<network-name>` : Όνομα του δικτύου που θα δημιουργηθεί.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση επιτυχίας ή αποτυχίας της δημιουργίας του δικτύου.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δημιουργήσεις νέο δίκτυο για χρήση από containers.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc network create my-network
```

Προβολή ενεργών δικτύων

Η εντολή `lxc network list` εμφανίζει μια λίστα με τα ενεργά δίκτυα στο σύστημα.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εμφανίζει μια λίστα με τα δίκτυα που είναι ενεργά στο σύστημα.

2. Ορίσματα και Ρυθμίσεις:

- Δεν απαιτούνται επιπλέον ορίσματα.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει τα διαθέσιμα δίκτυα μαζί με τις αντίστοιχες παραμέτρους τους.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να δεις ποια δίκτυα είναι ενεργά στο σύστημα.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc network list
```

Επεξεργασία δικτύου

Η εντολή `lxc network edit <network-name>` ανοίγει έναν επεξεργαστή για την τροποποίηση των ρυθμίσεων του δικτύου.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Επιτρέπει την επεξεργασία των ρυθμίσεων ενός δικτύου.

2. Ορίσματα και Ρυθμίσεις:

- `<network-name>` : Όνομα του δικτύου που θέλεις να επεξεργαστείς.

3. Πληροφορίες που Παρέχονται:

- Η εντολή ανοίγει έναν επεξεργαστή για την τροποποίηση του δικτύου.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να τροποποιήσεις τις ρυθμίσεις ενός δικτύου.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc network edit my-network
```

Διαγραφή δικτύου

Η εντολή `lxc network delete <network-name>` διαγράφει το συγκεκριμένο δίκτυο από το σύστημα.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Διαγράφει το καθορισμένο δίκτυο από το σύστημα.

2. Ορίσματα και Ρυθμίσεις:

- `<network-name>` : Όνομα του δικτύου που θέλεις να διαγράψεις.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση της διαγραφής του δικτύου.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να αφαιρέσεις ένα δίκτυο από το σύστημα.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc network delete my-network
```

Προσθήκη δικτύου σε container

Η εντολή `lxc network attach <network-name> <container-name>` προσθέτει ένα δίκτυο σε ένα container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Επισυνάπτει το καθορισμένο δίκτυο σε ένα container.

2. Ορίσματα και Ρυθμίσεις:

- `<network-name>` : Όνομα του δικτύου που θα προσαρτηθεί.
- `<container-name>` : Όνομα του container στο οποίο θα προσαρτηθεί το δίκτυο.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση της προσθήκης του δικτύου στο container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να συνδέσεις ένα δίκτυο σε ένα container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc network attach my-network container1
```

Ασφάλεια και Χρήστες

Προσθήκη χρήστη

Η εντολή `lxc config trust add <file.crt>` προσθέτει ένα πιστοποιητικό χρήστη για εμπιστοσύνη στο LXD.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Προσθέτει ένα νέο πιστοποιητικό χρήστη στην εμπιστοσύνη του LXD.

2. Ορίσματα και Ρυθμίσεις:

- `<file.crt>` : Αρχείο πιστοποιητικού χρήστη που θα προστεθεί.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση επιτυχίας ή αποτυχίας της προσθήκης του πιστοποιητικού.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να προσθέσεις νέο χρήστη στο LXD.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc config trust add user.crt
```

Κατάργηση χρήστη

Η εντολή `lxc config trust remove <fingerprint>` καταργεί την εμπιστοσύνη ενός χρήστη από το LXD.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Καταργεί την εμπιστοσύνη ενός χρήστη από το LXD μέσω του αναγνωριστικού δακτυλικού αποτυπώματος (fingerprint).

2. Ορίσματα και Ρυθμίσεις:

- `<fingerprint>` : Το δακτυλικό αποτύπωμα του χρήστη που θέλεις να καταργήσεις.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση της κατάργησης του χρήστη.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να καταργήσεις έναν χρήστη από το LXD.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc config trust remove 1234567890abcdef
```

Προβολή πιστοποιητικών χρηστών

Η εντολή `lxc config trust list` εμφανίζει μια λίστα με τα πιστοποιητικά χρηστών που είναι καταχωρημένα στο LXD.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εμφανίζει τα καταχωρημένα πιστοποιητικά χρηστών στο LXD.

2. Ορίσματα και Ρυθμίσεις:

- Δεν απαιτούνται επιπλέον ορίσματα.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει μια λίστα με τα πιστοποιητικά και τις πληροφορίες τους.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να δεις τα πιστοποιητικά των χρηστών που είναι αποδεκτά από το LXD.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc config trust list
```

Διαχείριση Πόρων

Περιορισμός μνήμης σε container

Η εντολή `lxc config set <container-name> limits.memory <value>` χρησιμοποιείται για να ορίσει τον περιορισμό μνήμης για ένα συγκεκριμένο container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Ορίζει τον περιορισμό μνήμης που μπορεί να χρησιμοποιήσει το container.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Όνομα του container που θέλεις να περιορίσεις τη μνήμη.
- `<value>` : Η μέγιστη ποσότητα μνήμης που μπορεί να χρησιμοποιήσει το container (π.χ., 1GB, 2GB).

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση της ρύθμισης του περιορισμού μνήμης.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να περιορίσεις τη μνήμη που μπορεί να καταναλώσει ένα container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc config set container1 limits.memory 2GB
```

Περιορισμός CPU σε container

Η εντολή `lxc config set <container-name> limits.cpu <value>` χρησιμοποιείται για να ορίσει τον περιορισμό CPU για το container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Ορίζει τον περιορισμό της CPU που μπορεί να χρησιμοποιήσει το container.

2. Ορίσματα και Ρυθμίσεις:

- **<container-name>** : Όνομα του container που θέλεις να περιορίσεις την CPU.
- **<value>** : Ο αριθμός των CPUs ή ποσοστό χρήσης της CPU που μπορεί να χρησιμοποιήσει το container.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση της ρύθμισης του περιορισμού της CPU.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να περιορίσεις τους πόρους CPU για ένα container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc config set container1 limits.cpu 2
```

Περιορισμός αποθηκευτικού χώρου

Η εντολή `lxc config device set <container-name> root size <value>` χρησιμοποιείται για να περιορίσει το μέγεθος του αποθηκευτικού χώρου του container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Ορίζει τον περιορισμό του αποθηκευτικού χώρου για το container.

2. Ορίσματα και Ρυθμίσεις:

- **<container-name>** : Όνομα του container που θέλεις να περιορίσεις τον αποθηκευτικό χώρο.
- **<value>** : Η μέγιστη ποσότητα αποθηκευτικού χώρου (π.χ., 10GB, 20GB).

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση της ρύθμισης του περιορισμού αποθηκευτικού χώρου.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να περιορίσεις τον αποθηκευτικό χώρο για το container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc config device set container1 root size 10GB
```

Προσθήκη επιπλέον αποθηκευτικού χώρου

Η εντολή `lxc config device add <container-name> <disk-name> disk source=<path>` προσθέτει επιπλέον αποθηκευτικό χώρο στο container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Προσθέτει επιπλέον αποθηκευτικό χώρο στο container μέσω ενός νέου δίσκου.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Όνομα του container στο οποίο θα προστεθεί ο δίσκος.
- `<disk-name>` : Όνομα του νέου δίσκου.
- `<path>` : Η πηγή του νέου δίσκου (π.χ., `/path/to/disk`).

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση προσθήκης του νέου δίσκου.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να προσθέσεις επιπλέον αποθηκευτικό χώρο στο container.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc config device add container1 extra-disk disk  
source=/mnt/data
```

Διαχείριση δικαιωμάτων πόρων

Η εντολή `lxc config set <container-name> security.privileged true/false` ελέγχει τα

δικαιώματα πόρων του container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Ρυθμίζει αν το container είναι «προνομιούχο» (privileged) ή όχι.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Όνομα του container.
- `true / false` : Ενεργοποίηση ή απενεργοποίηση των προνομίων για το container.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση της ρύθμισης ασφαλείας του container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να καθορίσεις αν το container θα έχει ή όχι προνομιακή πρόσβαση στους πόρους.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc config set container1 security.privileged true
```

Ενημερώσεις και Συντήρηση

Ενημέρωση LXD

Η εντολή `sudo snap refresh lxd` χρησιμοποιείται για να ενημερώσει την εγκατάσταση του LXD στην τελευταία διαθέσιμη έκδοση μέσω του Snap.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Ενημερώνει την εγκατάσταση του LXD στην τελευταία διαθέσιμη έκδοση από το Snap store.

2. Ορίσματα και Ρυθμίσεις:

- Δεν απαιτεί επιπλέον ορίσματα.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση της ενημέρωσης και την νέα έκδοση του LXD.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να διατηρήσεις το LXD ενημερωμένο με τις τελευταίες βελτιώσεις και διορθώσεις ασφαλείας.

5. Παράδειγμα Χρήσης:

```
root@server:~# sudo snap refresh lxd
```

Αναβάθμιση container

Η εντολή `lxc exec <container-name> -- apt update && apt upgrade` χρησιμοποιείται για να αναβαθμίσεις το σύστημα του container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Ενημερώνει και αναβαθμίζει τα πακέτα του container μέσω του `apt package manager`.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Όνομα του container που θέλεις να αναβαθμίσεις.

3. Πληροφορίες που Παρέχονται:

- Η εντολή εκτελεί την ενημέρωση (`apt update`) και την αναβάθμιση (`apt upgrade`) των πακέτων του container.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να διατηρήσεις τα containers ενημερωμένα με τις τελευταίες διορθώσεις και αναβαθμίσεις.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc exec container1 -- apt update && apt upgrade
```

Καθαρισμός μη χρησιμοποιούμενων images

Η εντολή `lxc image delete <image-name>` χρησιμοποιείται για να διαγράψεις ένα μη χρησιμοποιούμενο image από το LXD.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Διαγράφει το συγκεκριμένο image από την εγκατάσταση του LXD.

2. Ορίσματα και Ρυθμίσεις:

- `<image-name>` : Όνομα του image που θέλεις να διαγράψεις.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση διαγραφής του image.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να ελευθερώσεις χώρο και να καθαρίσεις μη χρησιμοποιούμενα images από το LXD.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc image delete ubuntu-20.04
```

Καθαρισμός στιγμιότυπων

Η εντολή `lxc delete <container-name>/<snapshot-name>` χρησιμοποιείται για να διαγράψεις ένα στιγμιότυπο από το container.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Διαγράφει το συγκεκριμένο στιγμιότυπο του container.

2. Ορίσματα και Ρυθμίσεις:

- `<container-name>` : Όνομα του container.
- `<snapshot-name>` : Όνομα του στιγμιότυπου που θέλεις να διαγράψεις.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση διαγραφής του στιγμιότυπου.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται για να διαγράψεις περιττά ή παλιά στιγμιότυπα και να ελευθερώσεις χώρο.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc delete container1/snapshot1
```

Διαχείριση Nodes (Cluster)

Δημιουργία cluster

Η εντολή `lxd init --cluster` χρησιμοποιείται για να δημιουργήσεις ένα νέο LXD cluster.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Δημιουργεί ένα νέο LXD cluster, επιτρέποντας τη διαχείριση πολλών nodes από ένα κεντρικό σημείο.

2. Ορίσματα και Ρυθμίσεις:

- Δεν απαιτεί επιπλέον ορίσματα, αλλά μπορεί να ζητηθούν πληροφορίες κατά τη διάρκεια της διαδικασίας αρχικοποίησης του cluster.

3. Πληροφορίες που Παρέχονται:

- Η εντολή αρχικοποιεί το cluster και διαμορφώνει τα απαιτούμενα στοιχεία για τη σύνδεση και τη διαχείριση των nodes.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να δημιουργήσεις ένα νέο LXD cluster για την κεντρική διαχείριση πολλών servers.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxd init --cluster
```

Προσθήκη node στο cluster

Η εντολή `lxc cluster add <node-name>` χρησιμοποιείται για να προσθέσεις ένα νέο node στο υπάρχον LXD cluster.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Προσθέτει ένα νέο node σε ένα ήδη υπάρχον LXD cluster.

2. Ορίσματα και Ρυθμίσεις:

- `<node-name>` : Όνομα του νέου node που θέλεις να προσθέσεις στο cluster.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση προσθήκης του node στο cluster.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να προσθέσεις ένα νέο node σε ένα υπάρχον cluster για να επεκτείνεις τη δυνατότητα του cluster.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc cluster add node2
```

Αφαίρεση node από cluster

Η εντολή `lxc cluster remove <node-name>` χρησιμοποιείται για να αφαιρέσεις ένα node από το LXD cluster.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Αφαιρεί ένα node από το LXD cluster.

2. Ορίσματα και Ρυθμίσεις:

- `<node-name>` : Όνομα του node που θέλεις να αφαιρέσεις από το cluster.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει την κατάσταση αφαίρεσης του node από το cluster.

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να αφαιρέσεις ένα node από το cluster για λόγους συντήρησης ή άλλες ανάγκες.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc cluster remove node2
```

Προβολή κατάστασης cluster

Η εντολή `lxc cluster list` χρησιμοποιείται για να δεις τη λίστα και την κατάσταση των nodes στο LXD cluster.

▼ Αναλυτικά

1. Λειτουργία της Εντολής:

- Εμφανίζει τη λίστα με τα nodes του LXD cluster και την τρέχουσα κατάστασή τους.

2. Ορίσματα και Ρυθμίσεις:

- Δεν απαιτεί άλλα ορίσματα.

3. Πληροφορίες που Παρέχονται:

- Η εντολή επιστρέφει τη λίστα των nodes στο cluster, μαζί με την κατάστασή τους (π.χ. online, offline).

4. Πότε Χρησιμοποιείται:

- Χρησιμοποιείται όταν θέλεις να ελέγξεις την κατάσταση και τη διαθεσιμότητα των

nodes του cluster.

5. Παράδειγμα Χρήσης:

```
root@server:~# lxc cluster list
```

Checklist (Junior Admin)

Παρακάτω είναι μια πιο σύντομη έκδοση του πίνακα, σχεδιασμένη ως checklist για έναν junior admin, εστιάζοντας στις βασικές εντολές. Αυτός ο πίνακας περιλαμβάνει τις πιο κοινές και απαραίτητες εντολές για τη διαχείριση του LXD.

Ενέργεια	Εντολή
Διαχείριση LXD	
Εγκατάσταση LXD	sudo apt install lxd
Εκκίνηση του LXD	lxd init
Επαλήθευση κατάστασης LXD	lxc-checkconfig
Διαχείριση Containers	
Δημιουργία container	lxc launch <image> <container-name>
Προβολή ενεργών containers	lxc list
Σύνδεση σε container	lxc exec <container-name> -- /bin/ bash
Διακοπή container	lxc stop <container-name>
Διακοπή όλων των container	lxc stop --all
Έναρξη container	lxc start <container-name>
Έναρξη όλων των container	lxc start --all
Κατάργηση container	lxc delete <container-name>

Ενέργεια	Εντολή
Δημιουργία στιγμιότυπου	<code>lxc snapshot <container-name> <snapshot-name></code>
Επαναφορά από στιγμιότυπο	<code>lxc restore <container-name> <snapshot-name></code>
Διαχείριση Δικτύων	
Δημιουργία προφίλ δικτύου	<code>lxc network create <network-name></code>
Προβολή ενεργών δικτύων	<code>lxc network list</code>
Διαχείριση Αρχείων	
Αντιγραφή αρχείου στο container	<code>lxc file push <local-path> <container>/<path></code>
Αντιγραφή αρχείου από το container	<code>lxc file pull <container>/<path> <local-path></code>
Ενημερώσεις και Συντήρηση	
Ενημέρωση LXD	<code>sudo snap refresh lxd</code>
Καθαρισμός μη χρησιμοποιούμενων images	<code>lxc image delete <image-name></code>

02 Virtualization

 [Ευαγγελία Κατσιαντώνη](#)  Λιγότερο από 1 λεπτό

Περιεχόμενα

- Virtualization | Ιστορία
- Ορισμός - Περιγραφή
- Software



Virtualization | Ιστορία

👤 Ευαγγελία Κατσιαντώνη 📅 8/12/24 🏢 Virtualization 📌 history ⌚ Περίπου 5 λεπτά

Ιστορική αναδρομή

Η τεχνολογία virtualization, με την έννοια που την ξέρουμε σήμερα, είναι μια σταδιακή εξέλιξη μέσω των εργασιών πολλών ερευνητών, εταιρειών και της παράλληλης εξέλιξης των υπολογιστικών συστημάτων. Η ιδέα ανήκει σε μια ερευνητική ομάδα της **IBM** πίσω στο μακρινό 1960.

Χρονικό εξέλιξης

60s:

Η ιδέα της δημιουργίας "εικονικού υλικού" υπήρχε από την δεκαετία του 1960, με τον υπολογιστή **IBM System/360** να φέρνει την έννοια του "**εικονικού μηχανήματος**" (virtual machine).

Στα τέλη της δεκαετίας του 1960 η **IBM** αναπτύσσει τον υπολογιστή **VM/370**, που επιτρέπει την εκτέλεση διαφόρων λειτουργικών συστημάτων σε ένα μόνο μηχάνημα.

70s:

Τη δεκαετία του 1970, το πρωτοπόρο ερευνητικό κέντρο **PARC** (Palo Alto Research Center) της **Xerox Corporation** στο Palo Alto της Καλιφόρνια, μελετά και εξελίσσει τεχνολογίες που έφεραν ριζικές αλλαγές στον τομέα της σύγχρονης τεχνολογίας των

υπολογιστών.

Αναπτύσσοντας τον **Alto**, έναν πρωτότυπο υπολογιστή με graphical user interface (GUI) και μια από τις πιο εξελιγμένες μορφές "**virtual memory**" από ό,τι υπήρχε τότε, επηρεάζει την μετέπειτα ανάπτυξη της virtualize.

80s:

Το 1987 ιδρύθηκε η VMware, γνωστή ως "VMware Incorporated". Η εταιρεία ξεκίνησε την ανάπτυξη λογισμικού virtualization για τον IBM PC και για άλλα συστήματα x86.

90s:

Το 1999 η VMware που ειδικεύεται σε τεχνολογίες virtualization, πρωτοκυκλοφόρησε το **VMware Workstation**.

Το VMware Workstation έφερε μια σειρά καινοτομιών, συμπεριλαμβανομένης της ικανότητας να τρέχει πολλαπλά λειτουργικά συστήματα ταυτόχρονα, να εξομοιώνει hardware και να δημιουργεί snapshots των VMs. Αυτό έδωσε στους χρήστες μια νέα ευελιξία και δυνατότητα να πραγματοποιούν πιο προηγμένα tasks στο υπολογιστή τους. Αυτή η έκδοση λογισμικού αποτέλεσε επανάσταση της εποχής, γιατί ήταν η πρώτη εμπορικά διαθέσιμη έκδοση λογισμικού virtualisation. Πριν από αυτήν υπήρχαν μόνο experimental ή ακαδημαϊκές εκδόσεις virtualisation που δεν ήταν εμπορικά διαθέσιμες.

Το προϊόν αυτό άνοιξε το δρόμο για τη μαζική χρήση της τεχνολογίας virtualization σε προσωπικούς υπολογιστές.

00s:

- Στο Πανεπιστήμιο του Cambridge το 2003 αναπτύχθηκε η τεχνολογία **Xen**.

Η τεχνολογία Xen είναι ένα open-source hypervisor, δηλαδή ένα λογισμικό που δημιουργεί virtual machines (VMs) πάνω σε ένα φυσικό hardware. Η τεχνολογία αυτή εξομοιώνει το hardware για να τρέξουν πολλαπλά λειτουργικά συστήματα στον ίδιο υπολογιστή, καθιστώντας έτσι πιο αποτελεσματική την χρήση των υπολογιστικών πόρων.

Η εταιρεία XenSource, η οποία αργότερα εξαγοράστηκε από την Citrix το 2008, ήταν υπεύθυνη για την εμπορική διάθεση και την περαιτέρω ανάπτυξή της.

- Η IBM, ο πρώτος διδάξας του Virtualization, το 2006 έρχεται ξανά στο προσκήνιο με την ανάπτυξη της τεχνολογίας **KVM** (Kernel-based Virtual Machine). Η τεχνολογία αποτελεί εικονικοποίηση υψηλής απόδοσης που ενσωματώνεται στο Linux kernel.

Το KVM έχει γίνει σήμερα η πιο δημοφιλής τεχνολογία virtualization για Linux συστήματα. Η ενσωμάτωσή του στο Linux kernel εξασφαλίζει άψογη συνεργασία με τα υφιστάμενα εργαλεία διαχείρισης και επιτρέπει την εκμετάλλευση όλων των δυνατοτήτων του λειτουργικού συστήματος. Η κοινότητα open-source και η υψηλή απόδοσή του το κάνουν ιδανικό για servers και cloud environments, με ποσοστό χρήσης περίπου 60% παγκοσμίως.

- Το 2007 από την **Innotek GmbH** κυκλοφόρησε για πρώτη φορά το **VirtualBox**, μια τεχνολογία που επιτρέπει στους χρήστες να δημιουργούν virtual machines (VMs) σε έναν υπολογιστή και να τρέχουν διαφορετικά λειτουργικά συστήματα ταυτόχρονα σ' αυτόν.

Η Innotek GmbH εξαγοράστηκε από την Sun Microsystems το 2008, και μετά από την εξαγορά της Sun από την Oracle το 2010, η Oracle αναλαμβάνει την ανάπτυξη και διανομή του Oracle VirtualBox.

Το Oracle VirtualBox λανσάρεται με τη σειρά του ορισμένες καινοτομίες σε σχέση με τις προηγούμενες τεχνολογίες εικονικοποίησης.

- **Cross-platform:** Το VirtualBox είναι διαθέσιμο για Windows, macOS και Linux συστήματα, οπότε μπορεί να χρησιμοποιηθεί σε ένα ευρύ φάσμα μηχανημάτων.
- **Open Source:** Το VirtualBox είναι open-source, οπότε μπορεί να κατέβει και να χρησιμοποιηθεί δωρεάν, κάτι που το κάνει εξαιρετικά ελκυστικό για συνήθεις χρήστες και developer.
- **Lightweight:** Το VirtualBox είναι μια ελαφριά εφαρμογή που δεν απαιτεί πολλούς πόρους για να τρέξει, κάτι που το κάνει ιδανικό για υπολογιστές με περιορισμένη μνήμη ή επεξεργαστική δύναμη.

Ωστόσο, η τεχνολογία του Oracle VirtualBox δεν έχει την ίδια αποτελεσματικότητα ή δυνατότητα για προηγμένα 'σενάρια ανάπτυξης' όπως οι KVM, Hyper-V, Xen και VMware. Χαρακτηριστικό παράδειγμα αποτελεί η αδυναμία για live migration ή dynamic memory management που προσφέρουν οι υπόλοιπες.

Συνολικά, το VirtualBox είναι μια εξαιρετική επιλογή για συνήθεις χρήστες που θέλουν να τρέξουν πολλαπλά λειτουργικά συστήματα στον ίδιο υπολογιστή. Για περισσότερα προηγμένα scenarios server virtualization, οι KVM, Hyper-V, Xen και VMware είναι καλύτερες επιλογές.

- Η **Microsoft** το 2008 ανέπτυξε και διέθεσε την τεχνολογία **Hyper-V**, για πρώτη φορά με τα Windows Server 2008. Η Microsoft με την Hyper-V ενσωματώνει μια σειρά καινοτομιών για την εποχή, που την καθιστούν δυνατό ανταγωνιστή στην κατηγορία του virtualization.
 - Direct Hardware Access: Η Hyper-V επιτρέπει στα VMs να έχουν άμεση πρόσβαση

σε hardware, όπως δίκτυα και αποθήκευση, μια καινοτομία που βελτιώνει την απόδοση και την ευελιξία.

- Live Migration: Η Hyper-V επιτρέπει την μεταφορά VMs από τον έναν server στον άλλο χωρίς να απαιτείται επανεκκίνηση, μια καινοτομία που βελτιώνει την διαθεσιμότητα και την διαχείριση του συστήματος.
- Dynamic Memory: Η Hyper-V αυτομάτως αλλάζει την ποσότητα μνήμης που χρησιμοποιεί μια VM βάσει της απαιτούμενης χρήσης, μια καινοτομία που βελτιώνει την αποτελεσματικότητα χρησιμοποίησης των υπολογιστικών πόρων.

Παράγοντες εξέλιξης

Παρατηρώντας το ιστορικό των εξελίξεων βλέπουμε ότι, από τη δεκαετία του 2000 εμφανίζονται αρκετές τεχνολογίες εικονικοποίησης. Οι κύριοι παράγοντες που ώθησαν την ανάπτυξή τους είναι:

- **Εντατικοποίηση πόρων**: Η ανάγκη για καλύτερη χρήση του υλικού και η εκτέλεση περισσότερων εφαρμογών ταυτόχρονα.
- **Ασφάλεια**: Η ανάγκη για ασφαλή εκτέλεση εφαρμογών και προστασία από κακόβουλο λογισμικό.
- **Ευελιξία**: Η εύκολη δημιουργία, διαγραφή και αναπαραγωγή εικονικών μηχανών προσφέρει πολύ μεγαλύτερη ευελιξία.

Σημείωση

Η τεχνολογία **virtualization** έχει αφήσει ένα τεράστιο σημάδι στον τομέα της πληροφορικής και όπως όλα δείχνουν, θα συνεχίσει να εξελίσσεται με νέες εφαρμογές.

Ορισμός - Περιγραφή

👤 Ευαγγελία Κατσιαντώνη 📅 8/3/24 🏢 Virtualization

🔗 Hyper-V, Oracle VM VirtualBox, VMware Workstation, XenServer ⌚ Περίπου 6 λεπτά

Ο όρος

Ο όρος **Virtualization**, εικονικοποίηση εν Ελληνιστί, είναι μια τεχνολογία που επιτρέπει σε ένα φυσικό κομμάτι υλικού (hardware) να υποστηρίζει την εκτέλεση πολλών λειτουργικών συστημάτων ταυτόχρονα, σαν να πρόκειται για ξεχωριστά συστήματα.

Virtualization ορίζεται η τεχνογνωστική της δημιουργίας εικονικών περιβαλλόντων, διάφορων εφαρμογών και χρήσεων, απομονωμένα μεταξύ τους, με αποτελεσματική διαχείριση των πόρων των υπολογιστών.

Κίνητρο ανάπτυξης

Η εικονικοποίηση έχει προκύψει από την ανάγκη να αξιοποιηθούν με τον πιο αποτελεσματικό τρόπο οι πόροι των υπολογιστών, να μείνει το κόστος λειτουργίας χαμηλό και να αυξηθεί η ασφάλεια των συστημάτων.

Αρχή λειτουργίας

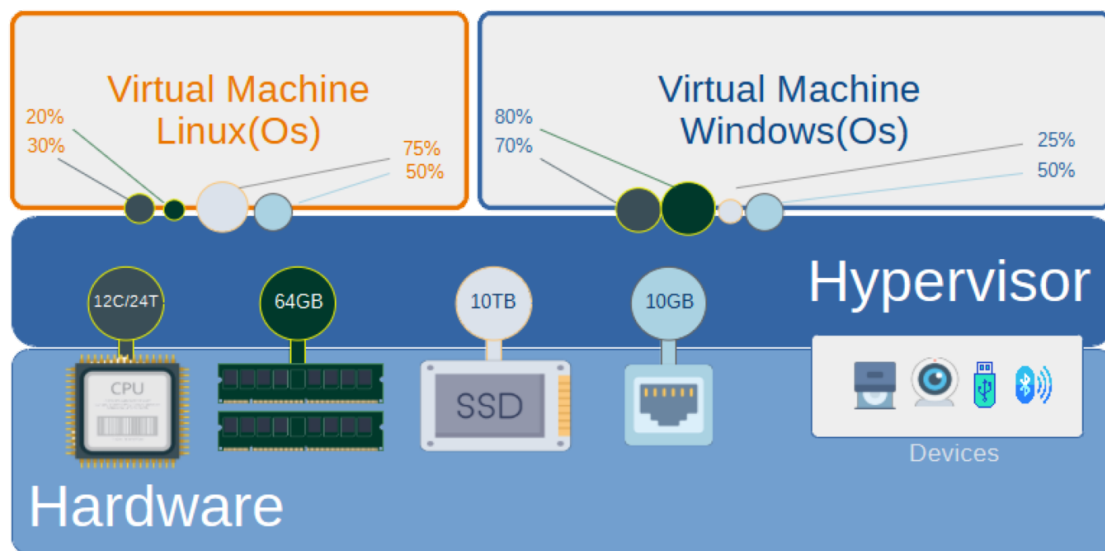
Η τεχνολογία της εικονικοποίησης (Virtualization) για να λειτουργήσει σε ένα σύστημα (hardware) απαιτεί ένα λογισμικό, το οποίο ονομάζεται **Hypervisor** (Υπερεπόπτης^[1]). Ο hypervisor δημιουργεί ένα εικονικό περιβάλλον (virtual machine, VM) πάνω στο φυσικό υλικό. Το εικονικό αυτό περιβάλλον (virtual machine) παρέχει ένα ολοκληρωμένο σύνολο πόρων, όπως CPU, μνήμη, δίσκοι, κάρτες δικτύου, κλπ., προσομοιώνοντας ένα πραγματικό σύστημα. Στο εικονικό περιβάλλον (virtual machine) μπορεί να εγκατασταθεί και να εκτελεστεί ένα λειτουργικό σύστημα, λειτουργώντας ανεξάρτητα από τα υπόλοιπα συστήματα.

Επιγραμματικά

1. **Hypervisor**: Ένα λογισμικό δημιουργεί το εικονικό περιβάλλον (virtual machine) πάνω στο φυσικό υλικό.

2. **VM:** Το εικονικό περιβάλλον παρέχει το σύνολο των πόρων (CPU, RAM, HD-SSD, LAN, DEVICES), προσομοιώνοντας ένα πραγματικό σύστημα.
3. **OS:** Ένα λειτουργικό σύστημα εγκαθιστάται και εκτελείται εντός του Virtual Machine, λειτουργώντας ανεξάρτητα από τα υπόλοιπα συστήματα.

Σχηματικά



Το παραπάνω σχήμα αποτελεί το βασικό μοντέλο εικονικοποίησης, στο οποίο ο Hypervisor:

- Διαχειρίζεται τον απευθείας έλεγχο του υλικού (hardware).
- Δημιουργεί ένα εικονικό περιβάλλον για κάθε OS.
- Διανέμει τους πόρους του υλικού στα εικονικά συστήματα.
- Παρέχει τις απαραίτητες υπηρεσίες για την απρόσκοπτη λειτουργία των εικονικών συστημάτων.

Τύποι hypervisors

Ανάλογα με τον τρόπο που εγκαθίσταται ο hypervisor στο υπολογιστικό σύστημα και την πρόσβαση που έχει στο hardware, τους διαχωρίζουμε σε δύο βασικούς τύπους:

1. Type 1 Hypervisors (Bare-Metal Hypervisors):

- Εγκαθίστανται άμεσα στο hardware του υπολογιστή, χωρίς να χρειάζεται ένα λειτουργικό σύστημα κάτω από αυτούς.
- Παρέχουν άμεση πρόσβαση στο hardware και προσφέρουν υψηλή απόδοση.

Γνωστοί Type 1 Hypervisors: VMware ESXi , XenServer , Microsoft Hyper-V Server , Proxmox VE , Oracle VM Server , Red Hat Virtualization , Citrix Hypervisor .

2. Type 2 Hypervisors (Hosted Hypervisors):

- Τρέχουν πάνω σε ένα λειτουργικό σύστημα, όπως Windows ή Linux.
- Είναι πιο εύκολα στη χρήση και κατάλληλα για δοκιμές και ανάπτυξη.

Γνωστοί Type 2 Hypervisors: VMware Workstation , Oracle VM VirtualBox , Parallels Desktop , Microsoft Hyper-V Server .

Θυμάμαι!

- **Type 1 Hypervisors:** Εγκαθίστανται άμεσα στο hardware και προσφέρουν υψηλή απόδοση για επαγγελματικές και επιχειρηματικές χρήσεις.
- **Type 2 Hypervisors:** Τρέχουν πάνω σε ένα λειτουργικό σύστημα και είναι πιο εύκολα στη χρήση για προσωπική χρήση και δοκιμές.

Επίπεδα εικονικοποίησης

Σε αντιστοιχία με τον διαχωρισμό στον τρόπο εγκατάστασης και πρόσβασης στο hardware των Hypervisors, διαχωρισμός γίνεται και στον τρόπο με τον οποίο υλοποιείται η εικονικοποίηση στο hardware και στο λογισμικό.

1. Virtualization Επίπεδου 1 (System virtualization):

- **Απευθείας Εικονικοποίηση (Direct Virtualization):** Ο hypervisor έχει άμεση πρόσβαση στο υλικό, επιτρέποντας την εκτέλεση διαφόρων λειτουργικών συστημάτων ταυτόχρονα.
- **Εικονικοποίηση Εντολών (Paravirtualization):** Το λειτουργικό σύστημα τροποποιείται ώστε να εκμεταλλεύεται τον hypervisor, κάνοντας την εικονικοποίηση πιο αποδοτική.

2. Virtualization Επίπεδου 2 (Process virtualization):

- **Virtualization Εφαρμογής:** Μια συγκεκριμένη εφαρμογή εκτελείται σε ένα εικονικό περιβάλλον, χωρίς να χρειάζεται να εγκατασταθεί ολόκληρο λειτουργικό σύστημα.
- **Εικονικοποίηση Δικτύου:** Η διακίνηση δεδομένων σε ένα δίκτυο εικονικοποιείται, επιτρέποντας την δημιουργία εικονικών δικτύων.

3. Virtualization Επίπεδου 3 (Storage virtualization):

- **Συγκέντρωση χώρου αποθήκευσης (Storage Aggregation):** Δημιουργεί ένα ενιαίο σημείο πρόσβασης σε πολλούς χώρους αποθήκευσης.
- **Εικονικοί δίσκοι (Virtual Disks):** Δημιουργεί εικονικούς δίσκους, ανεξάρτητους από τον φυσικό υλικό.

Θυμάμαι!

- Τα επίπεδα εικονικοποίησης αφορούν το επίπεδο στο οποίο γίνεται η εικονικοποίηση του hardware ή του λογισμικού.
- Κάθε επίπεδο προσφέρει διαφορετικά πλεονεκτήματα και χρησιμοποιείται για διαφορετικούς σκοπούς.
- Η τεχνολογία της εικονικοποίησης εξελίσσεται συνεχώς και νέα επίπεδα εικονικοποίησης αναπτύσσονται για να αντιμετωπίσουν τις σύγχρονες ανάγκες της τεχνολογίας.

Πλεονεκτήματα

Η ανάπτυξη της εικονικοποίησης έχει φέρει μια σειρά από πλεονεκτήματα που έχουν μεταμορφώσει τον τρόπο που χρησιμοποιούμε τους υπολογιστές. Η δυνατότητα να τρέχουν πολλά λειτουργικά συστήματα σε ένα μόνο κομμάτι hardware οδήγησε σε πιο αποτελεσματική χρήση των πόρων, μείωση του κόστους και αυξημένη ασφάλεια. Η εικονικοποίηση επιτρέπει την απομόνωση εφαρμογών, μειώνοντας τον κίνδυνο μόλυνσης από ιούς ή κακόβουλο λογισμικό. Η εύκολη δημιουργία και διαγραφή εικονικών μηχανών διευκολύνει την ανάπτυξη και τη διαχείριση εφαρμογών, ενώ η ελαστικότητα στην ανάπτυξη των πόρων επιτρέπει την προσαρμογή στις ανάγκες κάθε εφαρμογής.

Γιατί η τεχνολογία Virtualization έχει τόσο αλματώδη εξέλιξη;

- **Εντατικοποίηση πόρων:** Μπορούμε να εκμεταλλευτούμε καλύτερα το υλικό, εκτελώντας περισσότερες εργασίες ταυτόχρονα.
- **Ασφάλεια:** Η απομόνωση των εικονικών μηχανών μειώνει τον κίνδυνο μόλυνσης από ιούς ή κακόβουλες εφαρμογές.
- **Εύκολη διαχείριση:** Η virtualization διευκολύνει τη δημιουργία, τη διαγραφή, και την αναπαραγωγή εικονικών μηχανών.
- **Ελαστικότητα:** Μπορούμε εύκολα να αυξομειώνουμε τους πόρους μιας VM ανάλογα με τις ανάγκες.

Μειονεκτήματα

Ωστόσο, η εικονικοποίηση έχει και τα μειονεκτήματά της. Η δημιουργία μιας εικονικής μηχανής (VM) απαιτεί συνήθως σημαντικούς πόρους και η εκκίνησή της μπορεί να είναι σχετικά αργή.

Μια άλλη τεχνολογία, η containerization, η οποία έχει αναδυθεί ως μια νέα προσέγγιση στην διαχείριση εφαρμογών και υπηρεσιών, αντιμετωπίζει αυτά τα μειονεκτήματα. Η containerization επιτρέπει την εκτέλεση συγκεκριμένων εφαρμογών σε μια lightweight^[2] μονάδα που ονομάζεται container. Αυτά τα containers τρέχουν από το κοινό kernel του host systems και δεν απαιτούν την ανάπτυξη ολόκληρου του λειτουργικού συστήματος, οδηγώντας σε μικρότερο footprint και γρηγορότερες αποδόσεις.

Η Containerization περιγράφεται στην επόμενη ενότητα και είναι αυτή που χρησιμοποιείται ως επί το πλείστον στην ανάπτυξη της εργασίας σε αντίθεση με την Virtualization που παρέχει το VM για τη δημιουργία των containers.

Εφαρμογή

Η εικονικοποίηση, με την ικανότητά της να δημιουργεί εικονικά περιβάλλοντα, έχει βρει εφαρμογή σε ένα ευρύ φάσμα τομέων. Σήμερα, η εικονικοποίηση χρησιμοποιείται για την δημιουργία εικονικών servers και desktops, την ανάπτυξη και τη δοκιμή εφαρμογών, την εκτέλεση cloud services και την υποστήριξη εφαρμογών edge computing^[3]. Η εικονικοποίηση εξυπηρετεί την ανάγκη για αυξημένη αποδοτικότητα, εφαρμογή νέων τεχνολογιών και ευελιξία στην διαχείριση των IT υποδομών.

Οι προοπτικές για την εικονικοποίηση είναι πολύ υποσχόμενες, καθώς η τεχνολογία αναμένεται να εξελιχθεί με νέες λειτουργίες και να υποστηρίξει νέες εφαρμογές.

α την αν

Η εικονικοποίηση έχει μεταμορφώσει τον τρόπο που χρησιμοποιούμε τις IT υποδομές και θα συνεχίσει να διαδραματίζει κεντρικό ρόλο στην τεχνολογία του μέλλοντος.

Υποσημειώσεις

1. **Υπερεπόπτης:** Συνήθως, η λέξη "Hypervisor" χρησιμοποιείται και στα ελληνικά, χωρίς μετάφραση. ↵
2. **lightweight** συστήματα (ή μονάδες) υπολογιστών που έχουν σχεδιαστεί να είναι μικρά σε μέγεθος, ελαφριά σε βάρος και χαμηλής κατανάλωσης ενέργειας. ↵
3. **Edge computing:** Είναι μια υπολογιστική στρατηγική που φέρνει την υπολογιστική ισχύ πιο κοντά στους χρήστες, στα δεδομένα και στις συσκευές. Αντί να μεταφέρονται όλα τα δεδομένα σε ένα κεντρικό cloud data center, επεξεργάζονται τοπικά σε σημεία που βρίσκονται κοντά στην πηγή των δεδομένων. ↵

Software

👤 Ευαγγελία Κατσιαντώνη 📅 8/12/24 🗑️ Virtualization 💡 Software ⌚ Περίπου 3 λεπτά

Η εικονικοποίηση έχει εξελιχθεί σε μια πανταχού παρούσα τεχνολογία, ανοίγοντας δρόμους για ευέλικτες και αποτελεσματικές λύσεις για διάφορα περιβάλλοντα. Στον κόσμο της virtualization, υπάρχει πληθώρα εφαρμογών που προσφέρουν διαφορετικές λειτουργίες, ανάλογα με τις ανάγκες του χρήστη. Η επιλογή της σωστής λύσης εξαρτάται από το λειτουργικό σύστημα, την πλατφόρμα και το σκοπό για τον οποίο προορίζεται η εικονικοποίηση.

Windows

- **Hyper-V:** Ενσωματωμένη λύση virtualization από την Microsoft, διαθέσιμη σε Professional, Enterprise και Datacenter εκδόσεις των Windows.
- **VMware Workstation Pro:** Ένα ολιστικό πακέτο virtualization για Windows, με πλήρη λειτουργικότητα και προχωρημένες ρυθμίσεις.
- **VirtualBox:** Δωρεάν λύση virtualization με εύκολη χρήση και εξαιρετική υποστήριξη για πολλά λειτουργικά συστήματα.
- **Parallels Desktop:** Λύση virtualization για Windows με εστίαση στην εύκολη χρήση και υψηλή απόδοση.

macOS

- **Parallels Desktop:** Ιδανικό για εκτέλεση Windows σε macOS, με πλήρη ενσωμάτωση και υποστήριξη για όλες τις εκδόσεις του Windows.
- **VMware Fusion:** Ολιστική λύση virtualization με υψηλή απόδοση και προχωρημένες ρυθμίσεις, ιδανικό για επαγγελματική χρήση.
- **VirtualBox:** Δωρεάν λύση virtualization με ευκολία χρήσης και εξαιρετική υποστήριξη για πολλά λειτουργικά συστήματα.
- **LXD:** Open source daemon^[1] που προσφέρει συμβατότητα με το Docker και προσφέρει επιπλέον λειτουργίες για την διαχείριση LXC containers με γραφικό περιβάλλον διαχείρισης.

Linux

- **KVM (Kernel-based Virtual Machine):** Ενσωματωμένη λύση virtualization στο kernel του Linux, γνωστή για την απόδοσή της.
- **Xen:** Ισχυρή λύση virtualization με έμφαση στην απόδοση και την ασφάλεια, διαθέσιμη για διακομιστές και επιτραπέζιους υπολογιστές.
- **VirtualBox:** Δωρεάν λύση virtualization με εύκολη χρήση και εξαιρετική υποστήριξη για πολλά λειτουργικά συστήματα.

Bare-metal

- **XenServer:** Η Xen είναι μια από τις πρώτες τεχνολογίες virtualization ανοιχτού κώδικα. Ο XenServer είναι μια εμπορική έκδοση του Xen με υποστήριξη από την Citrix. Προσφέρει υψηλή απόδοση και αξιοπιστία, ιδανικός για διακομιστές με υψηλές απαιτήσεις.
- **Proxmox VE:** Είναι μια ολιστική λύση open source bare-metal^[2] hypervisor με ενσωματωμένη διαχείριση virtualization και containers. Προσφέρει πλήρη διαχείριση virtualized υποδομών μέσω μιας web διεπαφής, καθιστώντας τον ιδανικό για small to medium businesses.
- **OpenStack:** Είναι μια ανοιχτή πλατφόρμα cloud computing που περιλαμβάνει έναν bare-metal hypervisor (KVM). Ο OpenStack προσφέρει ένα πλήρες οικοσύστημα εργαλείων για τη διαχείριση cloud υποδομών.
- **KVM (Kernel-based Virtual Machine):** Είναι ένα ενσωματωμένο bare-metal hypervisor στο kernel του Linux. Παρόλο που δεν είναι μια ανεξάρτητη εφαρμογή virtualization, είναι η βάση για πολλές λύσεις bare-metal virtualization, όπως το Proxmox VE και το OpenStack.

Warning

- Είναι βασικό να διαβάσετε τις αναλύσεις και τις κριτικές για κάθε εφαρμογή virtualization πριν αποφασίσετε ποια να χρησιμοποιήσετε.
- Η επιλογή του open source bare-metal hypervisor εξαρτάται από τις ανάγκες σας, όπως η απόδοση, η ασφάλεια, η υποστήριξη λειτουργικών συστημάτων, οι διαθέσιμες λειτουργίες και το κόστος.

Υποσημειώσεις

1. **daemon:** Στο Linux, ο όρος "δαίμονας" (ή "daemon" στα αγγλικά) αναφέρεται σε ένα πρόγραμμα που εκτελείται στο παρασκήνιο, χωρίς να έχει άμεση αλληλεπίδραση με

τον χρήστη.

Αυτά τα προγράμματα, γνωστά και ως "background processes", εκτελούνται συνεχώς και παρέχουν βασικές υπηρεσίες στο σύστημα. ↵

2. **Bare-metal:** Ο όρος "bare-metal" στην τεχνολογία virtualization αναφέρεται σε ένα hypervisor που τρέχει άμεσα στο hardware του υπολογιστή, χωρίς να χρειάζεται ένα λειτουργικό σύστημα κάτω από αυτούς.

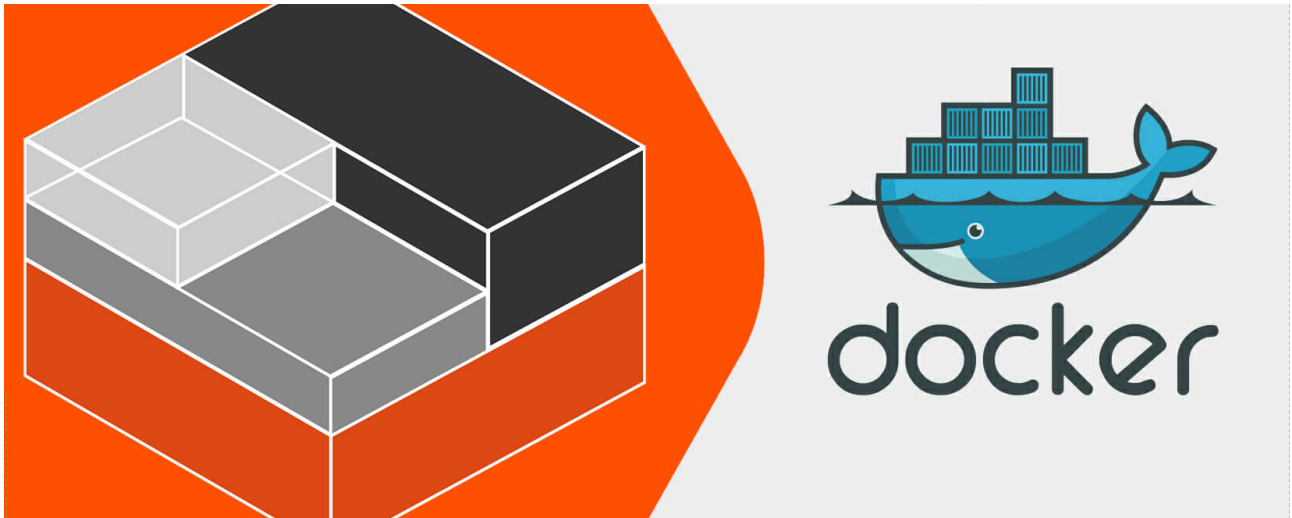
Σαν να "κολλάτε" τον hypervisor πάνω στο hardware χωρίς να χρειάζεται κάτι άλλο να "μεσολαβεί". ↵

03 Containerization

 [Ευαγγελία Κατσιαντώνη](#)  Λιγότερο από 1 λεπτό

Περιεχόμενα

- Containerization | Ιστορία
- Ορισμός - Περιγραφή
- Software
- Containerization Vs Virtualization
- LXC
- LXD



Containerization | Ιστορία

👤 Ευαγγελία Κατσιαντώνη 📅 8/12/24 🗂️ Containerization 🏷️ history ⌚ Περίπου 2 λεπτά

Ιστορική αναδρομή

Η ιστορική εξέλιξη της τεχνολογίας **containerization** είναι συναρπαστική. Έχει τις ρίζες της στις πρώτες προσπάθειες δημιουργίας μονωμένων περιβαλλόντων για εφαρμογές στο Unix. Επανάσταση στο χώρο αποτέλεσε το **Docker** και η ευρεία υιοθέτηση των containers σε όλους τους τομείς της τεχνολογίας εικονικοποίησης. Η τεχνολογία containerization έχει μεταμορφώσει τον τρόπο που δημιουργούνται και εκτελούνται οι εφαρμογές, προσφέροντας αποτελεσματικότητα, ευελιξία και αυξημένη ασφάλεια. Στο χρονικό αυτό, θα αναφερθούμε στα σημαντικά ορόσημα της εξέλιξης της τεχνολογίας containerization, από τις πρώτες προσπάθειες μέχρι τις συγκλονιστικές εξελίξεις του σήμερα.

Χρονικό εξέλιξης

1979

Το πρώτο σημαντικό βήμα έγινε με την εμφάνιση της Chroot (change root) στο Unix. Η Chroot, ένα απλό εργαλείο, μας επιτρέπει να αλλάξουμε το **root directory** του συστήματος για μια συγκεκριμένη

διαδικασία.

Αυτό σημαίνει ότι μια εφαρμογή βλέπει μόνο ένα μικρό μέρος του συστήματος files του host, δημιουργώντας ένα μικρό βαθμό μόνωσης.

1990s

Η δεκαετία του 1990 φέρνει την ανάπτυξη των FreeBSD jails και των Linux VServer. Αυτά τα συστήματα προσφέρουν μια πιο εξελιγμένη μορφή μονωμένων περιβαλλόντων για εφαρμογές, προσφέροντας μια πιο αποτελεσματική τεχνολογία containerization.

2000s

Η τεχνολογία virtualization κάνει τη μεγάλη της εμφάνιση με την ανάπτυξη hypervisors όπως το VMware ESXi και το Xen. Αυτές οι λύσεις επιτρέπουν την εκτέλεση πολλαπλών λειτουργικών συστημάτων πάνω σε ένα μοντέλο hardware, αλλά απαιτούν σημαντικούς πόρους.

2008

Η εμφάνιση του LXC (Linux Containers), ενσωματωμένο στο kernel του Linux, ανοίγει νέους ορίζοντες για την τεχνολογία containerization. Το LXC προσφέρει μια πιο ελαφριά και αποτελεσματική λύση σε σχέση με τα hypervisors, επιτρέποντας την εκτέλεση πολλαπλών εφαρμογών πάνω σε ένα μόνο σύστημα Linux χωρίς να χρειάζεται να τρέχει κάθε εφαρμογή σε ξεχωριστό virtual machine.

2013

Η εμφάνιση του Docker φέρνει μια επανάσταση στη χρήση containers. Το Docker διευκολύνει την δημιουργία, την εκτέλεση και τη διαχείριση containers με ένα πιο φιλικό προς τον χρήστη interface. Η δημιουργία του Docker Hub, μιας κεντρικής πλατφόρμας για την μοιρασιά και την κατάσταση εικόνων containers, έκανε την τεχνολογία ευρέως προσβάσιμη.

Το μέλλον

Η τεχνολογία containerization συνεχίζει να εξελίσσεται με νέες λειτουργίες, εφαρμογές και εργαλεία. Σήμερα, τα containers χρησιμοποιούνται ευρέως στο cloud computing, σε

DevOps workflows, και σε μικρο-υπηρεσίες, μεταμορφώνοντας τον τρόπο που δημιουργούνται και διαχειρίζονται οι εφαρμογές.

Η ιστορία της τεχνολογίας containerization αποδεικνύει ότι η αναζήτηση για πιο αποτελεσματικές και ευέλικτες λύσεις για την εκτέλεση εφαρμογών είναι μια συνεχής εξέλιξη με σημαντικές επιπτώσεις στη σύγχρονη τεχνολογία.

Το μέλλον, στον τρόπο διαχείρισης και απόδοσης της υπολογιστικής ισχύς, φαντάζει ευοίωνο.

Ορισμός - Περιγραφή

👤 Ευαγγελία Κατσιαντώνη 📅 8/12/24 🧩 Containerization 💡 LXD, LXC, Docker ⌚ Περίπου 5 λεπτά

Η containerization είναι μια τεχνολογία που έχει αναδυθεί τα τελευταία χρόνια και έχει μεταμορφώσει τον τρόπο με τον οποίο αναπτύσσονται, διαχειρίζονται και εκτελούνται οι εφαρμογές λογισμικού. Σε αυτό το άρθρο, θα εξερευνήσουμε την containerization, θα εξηγήσουμε τα κίνητρα για την ανάπτυξή της, θα αναλύσουμε την αρχή λειτουργίας της και θα εξετάσουμε τα πλεονεκτήματα και τα μειονεκτήματά της. Επιπλέον, θα συγκρίνουμε την containerization με την τεχνολογία virtualization και θα αναλύσουμε τις πρακτικές εφαρμογές της στην καθημερινή ζωή.

Ο όρος

Ο όρος **Containerization**, ενθυλάκωση ή εν Ελληνιστί, είναι μια τεχνολογία για την εκτέλεση εφαρμογών, όπου κάθε εφαρμογή τρέχει σε ένα μονωμένο περιβάλλον, γνωστό ως "**container**", το οποίο περιέχει όλα τα απαραίτητα components (κώδικας, δεδομένα, βιβλιοθήκες, interfaces, servers, databases) για να λειτουργήσει σωστά. Αυτά τα containers είναι ελαφριά και μεταφερόμενα, που σημαίνει ότι μπορούν να τρέξουν σε οποιοδήποτε σύστημα (hardware) με το ίδιο λειτουργικό σύστημα.

Η containerization είναι μια τεχνολογία που επιτρέπει σε εφαρμογές να τρέχουν σε ένα απομονωμένο περιβάλλον, γνωστό ως κοντέινερ, ανεξάρτητα από το υποκείμενο λειτουργικό σύστημα.

Σε αντίθεση με την Virtualization, η οποία δημιουργεί ένα πλήρες virtual machine για κάθε εφαρμογή, η Containerization διαφέρει γιατί χρησιμοποιεί έναν κοινό πυρήνα kernel για όλα τα containers, αξιοποιώντας τους πόρους πιο αποτελεσματικά.

Κίνητρο ανάπτυξης

Η ανάπτυξη της containerization οφείλεται σε μια σειρά από κίνητρα που σχετίζονται με τις ανάγκες του σύγχρονου περιβάλλοντος IT. Η μεγάλη ζήτηση για αποτελεσματικότητα πόρων, η ευελιξία του deployment, η αυξημένη ταχύτητα ανάπτυξης και η ασφάλεια οδήγησε στην εξέλιξη αυτής της τεχνολογίας. Η containerization επιτρέπει την εκτέλεση

πολλαπλών εφαρμογών σε έναν server, διασφαλίζοντας ομαλή διαχείριση και αναβάθμιση των applications με μείωση του hardware και του κόστους ενέργειας.

Αρχή λειτουργίας

Η containerization λειτουργεί με την δημιουργία μικρών, απομονωμένων περιβαλλόντων για εφαρμογές.

Η βάση της containerization είναι η εικόνα κοντέινερ (Image Container), η οποία περιλαμβάνει όλα τα απαραίτητα στοιχεία για την εκτέλεση μιας εφαρμογής, όπως ο ίδιος ο κώδικας, οι βιβλιοθήκες που χρειάζεται και οι ρυθμίσεις για την εκτέλεσή της.

Tips

Φανταστείτε την εικόνα σαν ένα "συμπιεσμένο" αρχείο που περιέχει όλα όσα χρειάζεται μια εφαρμογή για να λειτουργήσει.

Στη συνέχεια, ένα **runtime engine**, όπως το **Docker**, το **Lxc** ή το **Podman**, παίρνει την εικόνα και την ενθυλακώνει μέσα σε ένα ελαφρύ και απομονωμένο περιβάλλον, στο κοντέινερ. Το κοντέινερ αποτελεί ένα προστατευμένο "κουτί" για την εφαρμογή, εξασφαλίζοντας ότι δεν θα επηρεαστεί από άλλες εφαρμογές ή από το υποκείμενο λειτουργικό σύστημα.

Κάθε κοντέινερ τρέχει ανεξάρτητα, με αποτέλεσμα οι εφαρμογές να μην μοιράζονται πόρους μεταξύ τους (CPU, RAM, File System), ούτε να μπορούν να επηρεάσουν το ένα το άλλο. Η containerization εξασφαλίζει ότι μια εφαρμογή μπορεί να τρέξει με τον ίδιο τρόπο σε διαφορετικά λειτουργικά συστήματα, χωρίς να χρειάζεται να τροποποιηθεί ή να επανασχεδιαστεί.

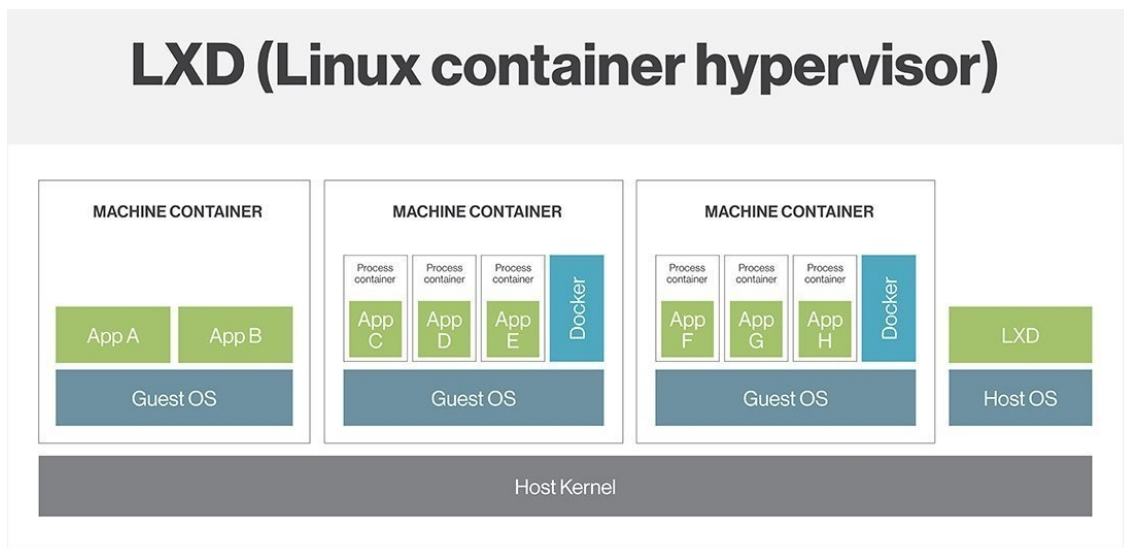
Containerization είναι μια τεχνολογία δημιουργίας, διαχείρισης και εκτέλεσης εφαρμογών με ευκολία, φορητότητα και αποδοτικότητα.

Επιγραμματικά

- Εικόνα Container (Image):** Η containerization ξεκινά με μια εικόνα container, η οποία είναι ένα στιγμιότυπο ενός περιβάλλοντος εκτέλεσης, συμπεριλαμβανομένου του λογισμικού, των βιβλιοθηκών και των ρυθμίσεων που χρειάζεται μια εφαρμογή για να λειτουργήσει.
- Κοντέινερ (Container):** Η εικόνα container χρησιμοποιείται για τη δημιουργία ενός κοντέινερ, το οποίο είναι ένα ελαφρύ και απομονωμένο περιβάλλον εκτέλεσης.

3. **Απομόνωση:** Τα κοντέινερ τρέχουν με απομόνωση, σημαίνει ότι οι εφαρμογές μέσα σε ένα κοντέινερ δεν μπορούν να επηρεάσουν άλλα κοντέινερ ή το υποκείμενο λειτουργικό σύστημα.
4. **Εκτέλεση:** Ο container runtime engine είναι υπεύθυνος για την εκτέλεση των κοντέινερ. Ενσωματώνει τους πόρους του συστήματος, όπως CPU, μνήμη και δίκτυο, και παρέχει ένα ασφαλές και σταθερό περιβάλλον για την εκτέλεση εφαρμογών.

Σχηματικά



Πλεονεκτήματα

- **Φορητότητα:** Τα κοντέινερ μπορούν να εκτελεστούν σε διαφορετικά λειτουργικά συστήματα και σε διάφορα περιβάλλοντα cloud (cloud environments).
- **Ασφάλεια:** Η απομόνωση των κοντέινερ εξασφαλίζει ότι οι εφαρμογές δεν επηρεάζουν η μία την άλλη.
- **Microservices:** Η containerization υποστηρίζει την ανάπτυξη και την εκτέλεση μικρο-υπηρεσιών.
- **Αποδοτικότητα:** Τα κοντέινερ είναι ελαφριά και γρήγορα, οπότε απαιτούν λιγότερους πόρους και χρόνο εκκίνησης.
- **Ευκολία διαχείρισης:** Η containerization απλοποιεί την διαχείριση των εφαρμογών, παρέχοντας ενιαία πακέτα για την εκτέλεση και την εφαρμογή αλλαγών.

Μειονεκτήματα

- **Ασυμβατότητα:** Η λειτουργία των containers εξαρτάται από το kernel του host

συστήματος.

- **Εξαρτήσεις:** Τα containers μπορεί να έχουν εξαρτήσεις από άλλα components, όπως libraries ή services.

Εφαρμογή

Η Containerization έχει γίνει ένα πανταχού παρόν εργαλείο για ομάδες ανάπτυξης, DevOps και εταιρείες cloud, εξυπηρετώντας αναγκαίες απαιτήσεις σε κάθε στάδιο του lifecycle^[1] της εφαρμογής.

Οι προγραμματιστές εκμεταλλεύονται την containerization για να δημιουργούν, να δοκιμάζουν και να αναπτύσσουν εφαρμογές με ευελιξία και φορητότητα, μείωση εξαρτήσεων και εγγυημένη συνεπή λειτουργία.

Οι ομάδες DevOps αξιοποιούν την containerization για να αυτοματοποιήσουν διαδικασίες διάθεσης, deployment, scaling και monitoring εφαρμογών, εξασφαλίζοντας ομαλή και αποτελεσματική διαχείριση των περιβαλλόντων ανάπτυξης και παραγωγής.

Επιπλέον, οι εταιρείες cloud εμπιστεύονται την containerization για να προσφέρουν υπηρεσίες cloud με μεγάλη ευελιξία και κλιμακωσιμότητα, συγκεντρώνοντας πολλά containers σε μια κοινή πλατφόρμα και εξασφαλίζοντας άψογη διαχείριση των πόρων.

Η containerization έχει μεταμορφώσει τον τρόπο με τον οποίο αναπτύσσονται και διαχειρίζονται εφαρμογές, προσφέροντας σημαντικά πλεονεκτήματα σε όλα τα στάδια του lifecycle μιας εφαρμογής.

Τεχνολογίες containerization

Για την δημιουργία, την ανάρτηση και τη διαχείριση containers υπάρχουν κορυφαίες τεχνολογίες, που παρέχουν ένα πλήρες οικοσύστημα. Ενδεικτικά αναφέρουμε τις:

- **Docker** : Η πιο δημοφιλής τεχνολογία containerization.
- **Kubernetes** : Ένα σύστημα orchestration για containers, που βοηθά στην αυτόματη διαχείριση μεγάλου αριθμού containers.
- **LXC (Linux Containers)** : Μια τεχνολογία containerization για Linux.

Η containerization έχει γίνει βασικό στοιχείο της μοντέρνας ανάπτυξης λογισμικού, παρέχοντας σημαντικά οφέλη σε όλους τους εμπλεκόμενους.

Υποσημειώσεις

1. **lifecycle**: Ο κύκλος ζωής μιας εφαρμογής, περιλαμβάνει όλα τα στάδια από την ιδέα έως και την απομάκρυνση της από την αγορά. Το lifecycle μπορεί να χωριστεί σε διάφορα στάδια, σχεδιασμός (Planning), ανάπτυξη (Development), συντήρηση (Maintenance) και απομάκρυνση (Retirement) μιας εφαρμογής. ↩

Software

👤 Ευαγγελία Κατσιαντώνη 📅 8/14/24 🏗️ Containerization 💻 Software ⌚ Περίπου 4 λεπτά

Εφαρμογές Containerization: Ένας Οδηγός για την Επιλογή

Η τεχνολογία containerization έχει εξελιχθεί σε κρίσιμο κομμάτι της σύγχρονης ανάπτυξης λογισμικού, προσφέροντας ευελιξία, αποτελεσματικότητα και εφαρμογή σε διαφορετικά περιβάλλοντα. Η επιλογή των κατάλληλων εργαλείων containerization είναι καθοριστικός παράγοντας για την επιτυχία οποιουδήποτε project. Αυτός ο οδηγός προσφέρει μια αναλυτική ματιά στις διαθέσιμες εφαρμογές, κατατάσσοντάς τις βάσει του λειτουργικού συστήματος στο οποίο απευθύνονται.

Linux

Docker

Η πιο δημοφιλής εφαρμογή containerization για Linux systems. Παρέχει ένα ολοκληρωμένο οικοσύστημα για την δημιουργία, την ανάρτηση και τη διαχείριση containers. Είναι ευρέως υιοθετημένη και διαθέτει μια τεράστια κοινότητα και μια πλούσια συλλογή εργαλείων και βιβλιοθηκών.

LXD

Ένα άλλο εργαλείο containerization για Linux, που προσφέρει μια εναλλακτική προσέγγιση στην διαχείριση containers. Ο LXD εξασφαλίζει υψηλότερο επίπεδο ασφάλειας και αποτελεσματικότητας, ιδανικά για περιπτώσεις όπου απαιτείται αυστηρή διαχείριση πόρων και μείωση των εξαρτήσεων.

Podman

Μια εναλλακτική λύση open-source για το Docker, που επιτρέπει την εκτέλεση containers χωρίς να απαιτείται daemon. Ο Podman προσφέρει υψηλή συμβατότητα με τα Docker images και διαθέτει ένα όλο και περισσότερο ευρύ οικοσύστημα.

Kubernetes

Ένα εργαλείο ενορχήστρωσης containers, το οποίο διαχειρίζεται έναν αριθμό containers σε ένα cluster. Είναι ιδανικό για εφαρμογές με υψηλή διαθεσιμότητα και απαιτήσεις κλιμάκωσης.

Windows

Docker for Windows

Μια έκδοση του Docker που τρέχει σε Windows systems. Παρέχει μια ομαλή διαδικασία ανάπτυξης containers για developers που χρησιμοποιούν Windows.

Windows Containers

Μια ενσωματωμένη λύση containerization για Windows Server, που προσφέρει συμβατότητα με εφαρμογές .NET και Java.

Kubernetes

Το Kubernetes μπορεί να τρέξει και σε Windows systems, προσφέροντας ενορχήστρωση containers σε υβριδικά περιβάλλοντα.

macOS

Docker for Mac

Μια έκδοση του Docker που τρέχει σε macOS systems, προσφέροντας μια ομαλή διαδικασία ανάπτυξης containers για developers που χρησιμοποιούν macOS.

Rancher

Ένα εργαλείο ενορχήστρωσης containers που τρέχει σε macOS systems, προσφέροντας

μια εύκολη λύση για την διαχείριση containers.

LXD vs Docker (Linux Systems)

Η επιλογή μεταξύ LXD και Docker σε ένα σύστημα Linux εξαρτάται από τις συγκεκριμένες απαιτήσεις που πρέπει να εξυπηρετηθούν.

LXD

- **Εικονικές μηχανές:** Το LXD δημιουργεί πλήρεις εικονικές μηχανές με βάση το Linux, οι οποίες έχουν τον δικό τους πυρήνα και διαχειρίζονται από τον LXD hypervisor.
- **Containers:** Το LXD χρησιμοποιεί το LXC (Linux Containers) για να δημιουργήσει ελαφριά, απομονωμένα containers.
- **Ισχυρή απομόνωση:** Η απομόνωση μεταξύ των εικονικών μηχανών LXD είναι ισχυρή, παρέχοντας αυξημένη ασφάλεια και σταθερότητα.
- **Υποστήριξη διαφορετικών διανομών:** Μπορείτε να εκτελέσετε διαφορετικές διανομές Linux σε εικονικές μηχανές LXD.
- **Καλύτερη συμβατότητα:** Το LXD υποστηρίζει καλύτερα τις εφαρμογές που απαιτούν άμεση πρόσβαση σε υλικό ή λειτουργικό σύστημα.

Docker

- **Ελαφριά εικονικοποίηση:** Ο Docker χρησιμοποιεί κοντέινερ, τα οποία μοιράζονται τον πυρήνα του υπολογιστή-ξενιστή, μειώνοντας τον πόρο που χρειάζεται για την εκτέλεση.
- **Γρήγορη εκκίνηση:** Τα κοντέινερ Docker ξεκινούν γρήγορα, καθιστώντας τα ιδανικά για μικρές υπηρεσίες ή για την ανάπτυξη.
- **Εύκολη διαχείριση:** Το Docker προσφέρει ένα ολοκληρωμένο σύνολο εργαλείων για την κατασκευή, την εκτέλεση και τη διαχείριση κοντέινερ.
- **Πολλά plugins:** Υπάρχουν πολλά plugins και εργαλεία για τον Docker, τα οποία επεκτείνουν τις δυνατότητες του.

Πίνακας

Χαρακτηριστικό	LXD	Docker
----------------	-----	--------

Χαρακτηριστικό	LXD	Docker
Τύπος εικονικοποίησης	Εικονικές μηχανές, Κοντέινερ	Κοντέινερ
Απομόνωση	Υψηλή	Μέτρια
Υποστήριξη διανομών	Πολλαπλές διανομές Linux	Κυρίως Linux, αλλά με περιορισμένη υποστήριξη Windows και MacOS
Εκκίνηση	Πιο αργή	Γρήγορη
Πόροι	Υψηλότερη χρήση πόρων	Χαμηλότερη χρήση πόρων
Συμβατότητα εφαρμογών	Υψηλή	Μέτρια
Διαχείριση	Λιγότερο φιλική προς το χρήστη	Φιλική προς το χρήστη
Plugins και εργαλεία	Λιγότερα plugins	Πολλά plugins
Ιδανικό για	Απαιτητικές εφαρμογές, ισχυρή ασφάλεια, υποστήριξη πολλαπλών διανομών	Μικρότερες εφαρμογές, γρήγορη ανάπτυξη, εύκολη διαχείριση
Παράδειγμα χρήσης	Εικονικοί servers, αποθήκευση δεδομένων, δοκιμή λογισμικού	Web servers, βάσεις δεδομένων, μικροσυμπυκνωμένες εφαρμογές

- **LXD:** Κατάλληλο για **πιο απαιτητικές εφαρμογές** που απαιτούν ισχυρή απομόνωση, υποστήριξη διαφορετικών διανομών Linux ή άμεση πρόσβαση σε υλικό.
- **Docker:** Κατάλληλο για **μικρότερες εφαρμογές, γρήγορη ανάπτυξη και εύκολη διαχείριση.**

Προσοχή

Πριν από την τελική επιλογή, λάβετε υπόψη τις **ανάγκες** σας, την **εμπειρία** σας, τα **πλεονεκτήματα** και τα **μειονεκτήματα** κάθε λύσης.

Συμπεράσματα

Η επιλογή των κατάλληλων εργαλείων containerization εξαρτάται από τις απαιτήσεις κάθε project. Το Docker είναι μια ισχυρή και ευρέως υιοθετημένη λύση για Linux, ενώ ο LXD προσφέρει υψηλότερο επίπεδο ασφάλειας και αποτελεσματικότητας. Το Kubernetes είναι ιδανικό για εφαρμογές με υψηλή διαθεσιμότητα και απαιτήσεις κλιμάκωσης. Για Windows systems, το Docker for Windows και τα Windows Containers προσφέρουν συμβατότητα με εφαρμογές .NET και Java. Τέλος, το Docker for Mac και το Rancher προσφέρουν λύσεις containerization για macOS systems.

Containerization Vs Virtualization

👤 Ευαγγελία Κατσιαντώνη 📅 8/12/24 🏷️ Containerization, Virtualization ⚔️ Vs ⌚ Περίπου 4 λεπτά

Στον σύγχρονο κόσμο της τεχνολογίας, η ανάγκη για ευελιξία και αποτελεσματικότητα στην ανάπτυξη και διαχείριση εφαρμογών είναι μεγαλύτερη από ποτέ. Η εικονικοποίηση (Virtualization) και η "κοντεϊνερποίηση" (Containerization) έχουν γίνει δύο κορυφαίες τεχνολογίες που προσφέρουν λύσεις σε αυτήν την ανάγκη. Αυτό το άρθρο θα εξετάσει τις βασικές διαφορές μεταξύ των δύο, ρίχνοντας φως στα πλεονεκτήματα και τα μειονεκτήματα κάθε προσέγγισης.

Virtualization

- **Αρχή λειτουργίας:** Η Virtualization δημιουργεί ένα "πλήρες" εικονικό περιβάλλον στον υπολογιστή. Δηλαδή, προσθέτει ένα επιπλέον επίπεδο λογισμικού που "εξαπατά" το λειτουργικό σύστημα και τις εφαρμογές κάνοντάς τις να πιστεύουν ότι τρέχουν σε ένα αυτόνομο σύστημα.
- **Απαιτούμενα:** Κάθε VM χρειάζεται το δικό της λειτουργικό σύστημα, driver και πολλούς πόρους.
- **Πλεονεκτήματα:**
- **Απομόνωση:** Κάθε VM είναι πλήρως απομονωμένη από άλλες VM και το host σύστημα. Αυτό εξασφαλίζει υψηλό επίπεδο ασφάλειας και σταθερότητας.
- **Ευελιξία:** Μπορείτε να τρέξετε διαφορετικά λειτουργικά συστήματα σε διαφορετικές VM.
- **Φορητότητα:** Είναι εύκολο να μεταφέρετε μια VM σε ένα διαφορετικό σύστημα.
- **Μειονεκτήματα:**
- **Χώρος:** Απαιτούν σημαντικό χώρο αποθήκευσης.
- **Επιδόσεις:** Είναι πιο αργές από τα containers.
- **Πόροι:** Χρειάζονται περισσότερους πόρους (CPU, μνήμη, disk space) για να τρέξουν.

Containerization

- **Αρχή λειτουργίας:** Η Containerization δημιουργεί ελαφριά, απομονωμένα περιβάλλοντα για εφαρμογές, χωρίς να χρειάζεται να τρέχουν πλήρη λειτουργικά συστήματα. Τα containers μοιράζονται το ίδιο kernel με το host σύστημα, εξασφαλίζοντας αυξημένη αποδοτικότητα και ελαφρότητα.
- **Απαιτούμενα:** Κάθε container χρειάζεται μια "image" που περιέχει τον κώδικα της

εφαρμογής και τις απαραίτητες εξαρτήσεις.

- **Πλεονεκτήματα:**
- **Ελαφρότητα:** Τα containers είναι πολύ ελαφριά και εκκινούν πολύ γρήγορα.
- **Αποδοτικότητα:** Χρησιμοποιούν λιγότερους πόρους από τις VM.
- **Φορητότητα:** Μπορείτε να τρέξετε containers σε οποιοδήποτε σύστημα με το ίδιο kernel.
- **Κλιμάκωση:** Είναι εύκολο να κλιμακώσετε containers για να αντιμετωπίσετε αύξηση στην ζήτηση.
- **Μειονεκτήματα:**
- **Ασφάλεια:** Λόγω της μοιραζόμενης kernel, τα containers είναι λιγότερο ασφαλή από τις VM.
- **Συμβατότητα:** Τα containers μπορούν να τρέξουν μόνο σε συστήματα με το ίδιο kernel.

Συνοπτικά

Feature	Virtualization	Containerization
Kernel	Separate kernel for each VM	Shared kernel with host
Resources	High resource usage	Low resource usage
Startup time	Slower	Faster
Portability	High	High
Isolation	High	Lower
Security	High	Lower
Complexity	High	Lower

Η επιλογή μεταξύ Virtualization και Containerization εξαρτάται από τις ανάγκες κάθε περίπτωσης.

Γνωρίζω

Η Virtualization προσφέρει υψηλό επίπεδο ασφάλειας και εφαρμοστικότητα.

Θυμάμαι

Η Containerization προσφέρει αυξημένη αποδοτικότητα, ελαφρότητα και προσαρμοστικότητα.

Επεξηγήσεις όρων

Kernel

- **Kernel:** Πυρήνας
- **Περιγραφή:** Ο πυρήνας (kernel) είναι η "καρδιά" ενός λειτουργικού συστήματος. Είναι υπεύθυνος για την διαχείριση των βασικών λειτουργιών του συστήματος, όπως:
- Διαχείριση πόρων: CPU, μνήμη, αποθηκευτικός χώρος.
- Διαχείριση διεργασιών.
- Διαχείριση δικτύου.
- Διαχείριση συσκευών.
- **Συγκριτικά:** Η Virtualization χρησιμοποιεί χωριστό kernel για κάθε VM, ενώ η Containerization μοιράζεται τον ίδιο kernel με το host σύστημα.

Resources

- **Resources:** Πόροι
- **Περιγραφή:** Οι πόροι ενός συστήματος είναι όλα όσα χρησιμοποιεί για να λειτουργήσει:
- CPU: Κεντρική μονάδα επεξεργασίας.
- Memory: Μνήμη RAM.
- Disk space: Χώρος αποθήκευσης.
- Network: Σύνδεση στο δίκτυο.
- **Συγκριτικά:** Η virtualization χρησιμοποιεί υψηλότερο χώρο πόρων λόγω του ξεχωριστού λειτουργικού συστήματος για κάθε VM, ενώ η containerization χρησιμοποιεί λιγότερους πόρους λόγω της μοιραζόμενης kernel.

Startup Time

- **Startup Time:** Χρόνος εκκίνησης.
- **Περιγραφή:** Αναφέρεται στον χρόνο που χρειάζεται ένα σύστημα, μια εφαρμογή ή ένα container για να φορτώσει και να ξεκινήσει να λειτουργεί.

- **Συγκριτικά:** Η containerization, συνήθως έχει πολύ μικρότερο startup time από τις Virtual Machines λόγω της μοιραζόμενης kernel.

Portability

- **Portability:** Φορητότητα
- **Περιγραφή:** Αναφέρεται στην ικανότητα ενός συστήματος ή μιας εφαρμογής να τρέξει σε διαφορετικά περιβάλλοντα χωρίς να χρειάζεται μεγάλες αλλαγές.
- **Συγκριτικά:** Η containerization και η virtualization προσφέρουν υψηλή φορητότητα αλλά με διαφορετικούς τρόπους. Η virtualization μπορεί να μεταφερθεί σε διαφορετικά συστήματα με διαφορετικά λειτουργικά συστήματα, ενώ η containerization χρειάζεται το ίδιο kernel.

Isolation

- **Isolation:** Απομόνωση.
- **Περιγραφή:** Αναφέρεται στην ικανότητα μιας εφαρμογής ή ενός συστήματος να τρέξει ανεξάρτητα από άλλα συστήματα ή εφαρμογές.
- **Συγκριτικά:** Η virtualization προσφέρει υψηλό επίπεδο απομόνωσης λόγω της χωριστής kernel. Η containerization προσφέρει λιγότερη απομόνωση, αλλά είναι αρκετή για να εξασφαλίσει ότι οι εφαρμογές δεν επηρεάζονται η μία την άλλη.

Security

- **Security:** Ασφάλεια.
- **Περιγραφή:** Αναφέρεται στην ικανότητα ενός συστήματος να προστατεύεται από μη επιτρεπόμενες πρόσβαση ή απειλές.
- **Συγκριτικά:** Η virtualization προσφέρει υψηλό επίπεδο ασφάλειας λόγω της χωριστής kernel και του εικονικού περιβάλλοντος. Η containerization είναι λιγότερο ασφαλής λόγω της μοιραζόμενης kernel, αλλά υπάρχουν μέτρα για να εξασφαλίσει την ασφάλεια των containers.

Complexity

- **Complexity:** Πολυπλοκότητα.
- **Περιγραφή:** Αναφέρεται στο επίπεδο δυσκολίας για να εγκατασταθεί, να διαχειριστεί ή να χρησιμοποιηθεί ένα σύστημα ή μια εφαρμογή.
- **Συγκριτικά:** Η virtualization είναι συνήθως πιο πολύπλοκη από την containerization. Η

containerization είναι πιο ευκολότερη στην εγκατάσταση και στη διαχείριση.

Separate kernel for each VM

- **Separate kernel for each VM:** Χωριστός kernel για κάθε VM.
- **Περιγραφή:** Αυτό σημαίνει ότι κάθε virtual machine (VM) τρέχει το δικό της λειτουργικό σύστημα και το δικό της kernel.
- **Συγκριτικά:** Αυτό δημιουργεί υψηλό επίπεδο απομόνωσης και εξασφαλίζει ότι οι εφαρμογές δεν επηρεάζονται η μία την άλλη.

Shared kernel with host

- **Shared kernel with host:** Μοιραζόμενος kernel με το host σύστημα.
- **Περιγραφή:** Αυτό σημαίνει ότι τα containers μοιράζονται τον ίδιο kernel με το host σύστημα.
- **Συγκριτικά:** Αυτό εξασφαλίζει αυξημένη αποδοτικότητα και μικρότερη χρήση πόρων σε σχέση με την virtualization.

High resource usage

- **High resource usage:** Υψηλή χρήση πόρων.
- **Περιγραφή:** Αυτό αναφέρεται στην ικανότητα ενός συστήματος ή μιας εφαρμογής να χρησιμοποιεί πολλούς πόρους (CPU, μνήμη, disk space).
- **Συγκριτικά:** Η virtualization χρησιμοποιεί υψηλή χρήση πόρων λόγω του ξεχωριστού λειτουργικού συστήματος για κάθε VM.

Εισαγωγή

Το LXC (Linux Containers) αποτελεί ένα σύστημα για ελαφριά εικονικοποίηση σε περιβάλλοντα Linux. Επιτρέπει την εκτέλεση πολλαπλών, απομονωμένων περιβαλλόντων εφαρμογών ("κοντέινερ") σε ένα μόνο φυσικό σύστημα, μοιράζοντας τον πυρήνα του λειτουργικού συστήματος. Η ελαφρότητά του, η ταχύτητα εκκίνησης, η χαμηλή κατανάλωση πόρων και η ευελιξία του LXC το καθιστούν ιδανικό για μια ποικιλία εφαρμογών, από την ανάπτυξη λογισμικού και την δοκιμή έως την φιλοξενία εφαρμογών και την διαχείριση υποδομών.

LXC: Σύστημα Ελαφριάς Εικονικοποίησης σε Linux

Τεχνικά Χαρακτηριστικά

Το LXC, βασισμένο στο "cgroups" και "namespaces" του πυρήνα Linux, δημιουργεί κοντέινερ που είναι απομονωμένα σε επίπεδο:

- **Δικτύου:** Κάθε κοντέινερ έχει τη δική του διαμόρφωση δικτύου, επιτρέποντας την δημιουργία ξεχωριστών εικονικών δικτύων.
- **Πόρων:** Κάθε κοντέινερ μπορεί να περιορίζεται ή να ελέγχεται σε σχέση με την χρήση CPU, μνήμης, αποθήκευσης και άλλων πόρων.
- **Ιδιότητες:** Κάθε κοντέινερ έχει τις δικές του ιδιότητες, όπως όνομα, ομάδα και διαμόρφωση, προσφέροντας αυξημένη οργάνωση και διαχείριση.
- **Προσωπικές συσκευές:** Η διαχείριση των συσκευών του κοντέινερ γίνεται με βάση τις ανάγκες κάθε περιβάλλοντος, επιτρέποντας την πρόσβαση σε συγκεκριμένες συσκευές.

Η χρήση των "cgroups" επιτρέπει την διαχείριση της χρήσης πόρων και την ισοκατανομή τους μεταξύ των κοντέινερ, ενώ τα "namespaces" δημιουργούν ξεχωριστά περιβάλλοντα για κάθε κοντέινερ.

Πλεονεκτήματα LXC

- **Ελαφρύτητα:** Τα κοντέινερ LXC είναι ελαφριά, καθώς μοιράζονται τον ίδιο πυρήνα λειτουργικού συστήματος με τον υπολογιστή-ξενιστή.
- **Ταχύτητα εκκίνησης:** Τα κοντέινερ LXC ξεκινούν πολύ γρήγορα σε σύγκριση με τις εικονικές μηχανές.
- **Χαμηλή κατανάλωση πόρων:** Η χαμηλή χρήση πόρων καθιστά το LXC ιδανικό για συστήματα με περιορισμένους πόρους.
- **Ευελιξία:** Η ευελιξία του LXC επιτρέπει την εύκολη δημιουργία, διαχείριση και κλιμάκωση κοντέινερ.
- **Ασφάλεια:** Η απομόνωση μεταξύ των κοντέινερ αυξάνει την ασφάλεια του συστήματος.
- **Εξομοίωση:** Η δυνατότητα δημιουργίας κοντέινερ με διαφορετικές διανομές Linux, επιτρέπει την εύκολη εξομοίωση διαφορετικών περιβαλλόντων.

Χρήσεις LXC

- **Ανάπτυξη λογισμικού:** Το LXC προσφέρει ένα σταθερό περιβάλλον ανάπτυξης, επιτρέποντας στους προγραμματιστές να δημιουργήσουν και να δοκιμάσουν κώδικα χωρίς να επηρεάζουν άλλα περιβάλλοντα.
- **Δοκιμή:** Το LXC είναι ιδανικό για την δοκιμή λογισμικού σε διαφορετικά περιβάλλοντα, εξομοιώνοντας διαφορετικές διανομές Linux.
- **Φιλοξενία εφαρμογών:** Η ελαφρότητα και η ευελιξία του LXC το καθιστούν ιδανικό για φιλοξενία εφαρμογών σε ένα μόνο σύστημα, μειώνοντας τους πόρους και το κόστος.
- **Διαχείριση υποδομών:** Το LXC μπορεί να χρησιμοποιηθεί για την διαχείριση υποδομών, επιτρέποντας την δημιουργία και την διαχείριση κοντέινερ για διάφορες υπηρεσίες.

LXC vs Docker

Το LXC αποτελεί μια πιο "χαμηλού επιπέδου" τεχνολογία, ενώ το Docker, βασισμένο στο LXC, προσφέρει ένα πιο φιλικό προς το χρήστη περιβάλλον για την κατασκευή, εκτέλεση και διαχείριση κοντέινερ. Το Docker, με τα προ-built images και τα εργαλεία automation, απλοποιεί τη διαδικασία δημιουργίας και χρήσης κοντέινερ, ενώ το LXC προσφέρει μεγαλύτερη ευελιξία και έλεγχο.

Συμπεράσματα

Το LXC αποτελεί ένα ισχυρό και ελαφρύ πλαίσιο για εικονικοποίηση, προσφέροντας μια ποικιλία πλεονεκτημάτων για διαφορετικά use cases. Η ελαφρότητα, η ταχύτητα, η χαμηλή χρήση πόρων, η ευελιξία και η ασφάλεια του LXC το καθιστούν ιδανικό για την ανάπτυξη, την δοκιμή, την φιλοξενία και τη διαχείριση εφαρμογών σε περιβάλλοντα Linux.

Παρατηρήσεις

- Το LXC είναι μια τεχνολογία που εξελίσσεται συνεχώς, με νέες δυνατότητες και βελτιώσεις.
- Υπάρχουν πολλά εργαλεία που μπορούν να χρησιμοποιηθούν μαζί με το LXC, όπως το LXD, για να διευκολύνουν τη διαχείριση των κοντέινερ.
- Η επιλογή μεταξύ LXC και Docker εξαρτάται από τις συγκεκριμένες ανάγκες του project.

Σημαντικοί Συνδεσμοί

- [LXC Website](#)
- [LXC Documentation](#)
- [Ubuntu LXC Documentation](#)
- [Docker Website](#)

LXD

Ευαγγελία Κατσιαντώνη 8/12/24 Containerization LXD Περίπου 4 λεπτά

Εισαγωγή

Το LXD, που σημαίνει Linux Container Daemon, είναι ένα σύγχρονο και εξελιγμένο εργαλείο για τη διαχείριση κοντέινερ Linux. Βασίζεται στο LXC (Linux Containers) και προσφέρει ένα ολοκληρωμένο σύνολο λειτουργιών για τη δημιουργία, τη διαχείριση και την κλιμάκωση κοντέινερ σε περιβάλλοντα Linux. Η ευελιξία του LXD, η υψηλή απόδοση, η ασφάλεια και η φιλική προς το χρήστη διεπαφή το καθιστούν ιδανικό εργαλείο για μια ποικιλία εφαρμογών, από την ανάπτυξη λογισμικού και την φιλοξενία εφαρμογών έως τη διαχείριση υποδομών.

LXD: Ένα Πλαίσιο Διαχείρισης Κοντέινερ για Υψηλή Απόδοση και Ασφάλεια

Τεχνικά Χαρακτηριστικά

Το LXD, με βάση το LXC, προσφέρει μια σειρά από σημαντικά χαρακτηριστικά που το διαφοροποιούν. Η αρχιτεκτονική του LXD βασίζεται σε hypervisor, επιτρέποντας την δημιουργία εικονικών περιβαλλόντων (instances) για κάθε κοντέινερ, με αποτέλεσμα κάθε instance να είναι πλήρως απομονωμένο, προσφέροντας ασφάλεια και σταθερότητα. Η διαχείριση πόρων, επιτρέπει τον έλεγχο των πόρων που χρησιμοποιούνται από κάθε instance, δίδοντας τη δυνατότητα να καθορίσετε τα όρια χρήσης CPU, μνήμης, αποθήκευσης και άλλων πόρων για κάθε κοντέινερ.

- Hypervisor:** Το LXD λειτουργεί ως hypervisor, δημιουργώντας εικονικά περιβάλλοντα (instances) για κάθε κοντέινερ. Αυτά τα instances είναι πλήρως απομονωμένα, προσφέροντας ασφάλεια και σταθερότητα.
- Διαχείριση Πόρων:** Το LXD επιτρέπει τον έλεγχο των πόρων που χρησιμοποιούνται από κάθε instance. Μπορείτε να καθορίσετε τα όρια χρήσης CPU, μνήμης, αποθήκευσης και άλλων πόρων για κάθε κοντέινερ.
- Δικτύωση:** Το LXD υποστηρίζει εξελιγμένες δικτυακές διαμορφώσεις, επιτρέποντας τη δημιουργία εικονικών δικτύων (virtual networks), την σύνδεση κοντέινερ σε διαφορετικά δίκτυα και την εφαρμογή πολιτικών ασφαλείας δικτύου.
- Αποθήκευση:** Το LXD διαχειρίζεται την αποθήκευση των κοντέινερ με ευελιξία. Υποστηρίζει διάφορα είδη αποθήκευσης, όπως το ZFS και το Btrfs, και προσφέρει δυνατότητες δημιουργίας αντιγράφων ασφαλείας (backups) και αποκατάστασης.

(restores).

- **API και Εργαλεία:** Το LXD παρέχει ένα ισχυρό API για αυτοματοποίηση της διαχείρισης κοντέινερ, καθώς και εργαλεία γραμμής εντολών (CLI) και γραφικά περιβάλλοντα (GUI) για εύκολη χρήση.

Πλεονεκτήματα LXD

Το LXD, χάρη στην υψηλή απόδοση, την ασφάλεια, την ευελιξία και την φιλικότητα προς το χρήστη, αποτελεί ένα ισχυρό εργαλείο για τη διαχείριση κοντέινερ σε μια ποικιλία περιβάλλοντων. Η ευελιξία του LXD, η δυνατότητα κλιμάκωσης, η ολοκληρωμένη διαχείριση και η δυνατότητα αυτοματοποίησης, το καθιστούν ιδανικό για σύγχρονα περιβάλλοντα ανάπτυξης, φιλοξενίας και διαχείρισης εφαρμογών.

- **Υψηλή Απόδοση:** Η αρχιτεκτονική του LXD βασίζεται σε hypervisor, επιτρέποντας υψηλή απόδοση και χαμηλή αδρανοποίηση.
- **Ασφάλεια:** Η πλήρης απομόνωση μεταξύ των instances εξασφαλίζει υψηλό επίπεδο ασφάλειας.
- **Ευελιξία:** Το LXD υποστηρίζει μια ποικιλία δικτυακών διαμορφώσεων, επιλογών αποθήκευσης και δυνατοτήτων διαχείρισης.
- **Κλιμάκωση:** Το LXD υποστηρίζει εύκολη κλιμάκωση, επιτρέποντας την προσθήκη νέων instances σε ένα cluster χωρίς διακοπή λειτουργίας.
- **Χρηστικότητα:** Το LXD παρέχει ένα φιλικό προς το χρήστη CLI και API, καθώς και εργαλεία διαχείρισης.
- **Συνεργασία:** Το LXD είναι ένα open source project, με μεγάλη κοινότητα και ενεργή ανάπτυξη.

Χρήσεις LXD

Το LXD, με σταθερό και ασφαλές περιβάλλον, είναι ιδανικό για ανάπτυξη και δοκιμή λογισμικού. Επιπλέον, η υψηλή απόδοση και ασφάλεια το καθιστούν ιδανικό για φιλοξενία εφαρμογών σε cloud περιβάλλοντα.

- **Ανάπτυξη Λογισμικού:** Το LXD παρέχει ένα σταθερό και ασφαλές περιβάλλον για την ανάπτυξη και την δοκιμή λογισμικού.
- **Φιλοξενία Εφαρμογών:** Το LXD είναι ιδανικό για φιλοξενία εφαρμογών σε cloud περιβάλλοντα, με υψηλή απόδοση και ασφάλεια.
- **Διαχείριση Υποδομών:** Το LXD μπορεί να χρησιμοποιηθεί για τη διαχείριση υποδομών, με δυνατότητες αυτοματοποίησης και κλιμάκωσης.

- **Εκπαίδευση και Έρευνα:** Το LXD προσφέρει ένα ιδανικό περιβάλλον για εκπαιδευτικούς σκοπούς και έρευνα σε θέματα εικονικοποίησης και διαχείρισης κοντέινερ.

LXD vs Docker

Το LXD και το Docker είναι δύο δημοφιλή εργαλεία για τη διαχείριση κοντέινερ. Το LXD εστιάζει στην υψηλή απόδοση και ασφάλεια, ενώ το Docker προσφέρει ένα πιο φιλικό προς το χρήστη περιβάλλον για την δημιουργία και εκτέλεση κοντέινερ. Η επιλογή μεταξύ LXD και Docker εξαρτάται από τις συγκεκριμένες ανάγκες του project.

Συμπεράσματα

Το LXD, με την εστίαση στην υψηλή απόδοση, την ασφάλεια, την ευελιξία και τη χρηστικότητα, αποτελεί ένα ισχυρό εργαλείο για τη διαχείριση κοντέινερ σε μια ποικιλία περιβάλλοντων. Η ευελιξία του, η δυνατότητα κλιμάκωσης, η ολοκληρωμένη διαχείριση και η δυνατότητα αυτοματοποίησης, το καθιστούν ιδανικό για σύγχρονα περιβάλλοντα ανάπτυξης, φιλοξενίας και διαχείρισης εφαρμογών.

Παρατηρήσεις

- Το LXD είναι μια τεχνολογία που εξελίσσεται συνεχώς, με νέες δυνατότητες και βελτιώσεις.
- Το LXD είναι ένα open source project, με μεγάλη κοινότητα και ενεργή ανάπτυξη.
- Η επιλογή μεταξύ LXD και Docker εξαρτάται από τις συγκεκριμένες ανάγκες του project.

Σημαντικοί Συνδεσμοί

- [LXD Website](#)
- [LXD Documentation](#)
- [Docker Website](#)