Bahir Dar UniversityBahir Dar Institute of Technology (Bit)

Faculty of ComputingDepartment of information

Individual Assignment _Omnios- IS_BIruktawit-Getaachew_ Adamu_bdu01601165.

Introduction

Background

OMNIOS : is an enterprise-grade, open-source Unix-like operating system based on the illumos kernel, which originated from OpenSolaris. After Oracle discontinued

OpenSolaris, the illumos community continued its development, and OmniOS emerged as a stable, server-focused distribution designed for reliability, performance, and long-term maintainability.

An Operating System (OS) is system software that manages computer hardware and provides services to applications. OmniOS is an enterprise-grade UNIX operating system based on illumos, derived from OpenSolaris.

It is widely used for servers, storage systems, and advanced networking.

Virtualization allows OmniOS to be installed safely on standard hardware (such as Gigabyte or Asus systems) using tools like Oracle VM VirtualBox, **without affecting the host OS.**

Motivation

OmniOS is an open-source enterprise-grade Unix-like operating system based on the illumos kernel — itself a community-driven continuation of OpenSolaris after Oracle discontinued it**.** It focuses on stability, performance, and reliability especially for servers, storage, and infrastructure workloadsTo gain practical experience installing a UNIX-basedOSTo understand Installation Oracle virtual box

Objectives

The objectives of this installation are: To install OmniOS in a virtual environmentTo understand the installation process and system configuration

To gain practical experience with Unix-based server operating systems

To study filesystem support, especially ZFS

To understand virtualization concepts in modern operating systems

Objectives of Using OmniOS

The main objectives of studying and installing OmniOS are:

To understand enterprise-grade operating systems

To gain knowledge of how enterprise-focused Unix-like operating systems are designed for stability, reliability, and long-term operation.

To install and configure OmniOS in a virtual environment

To practically install OmniOS using virtualization tools such as VMware Workstation or Oracle VM VirtualBox and understand virtual hardware configuration.

To learn advanced filesystem concepts (ZFS)

To study the features of ZFS, including data integrity, snapshots, volume management, and scalability, and understand why it is suitable for servers.

To understand Solaris/illumos-based operating systems

To learn the historical evolution from Solaris → OpenSolaris → illumos → OmniOS and its impact on modern server operating systems.

To gain hands-on experience with system administration tasks

To perform basic administrative tasks such as user creation, package management, service control, and system monitoring.

To analyze system performance and reliability

To observe OmniOS behavior under load and understand how it manages CPU, memory, storage, and I/O efficiently.

To understand virtualization concepts in modern operating systems

To understand how virtualization works, why it is used, and how OmniOS operates as a guest OS in a virtualized environment.

To compare OmniOS with other operating systems

To evaluate OmniOS against Linux, Windows, and BSD systems in terms of filesystem support, performance, and use cases.

To develop confidence in technical decision-making

To justify technical choices such as selecting ZFS over other filesystems and choosing OmniOS for server and infrastructure workloads.

To prepare for real-world server and infrastructure environments

To build foundational skills applicable to data centers, cloud infrastructure, storage systems, and enterprise IT environments.

Requirements for OmniOS

1. Hardware RequirementsComponen Minimum RequirementRecommendedCPU64-bit x86 processor

Multi-core 64-bit CPUVirtualization Support

Intel VT-x / AMD-VMust be enabled

RAM2 GB4–8 GB or more

Storage :20 GB HDD/SSD 40 GB+ (ZFS friendly)Network Ethernet / Virtual NIC

Intel NIC preferredArchitecturex86_64 only Required

 OmniOS has limited support for modern consumer hardware, so virtual machines are strongly recommended.

Software Requirements for OmniOS

To install and run OmniOS successfully, several software components are required. First, the OmniOS ISO image is needed. This is the official installation

file used to install the operating system. It should be downloaded from the official OmniOS project source to ensure stability and security.

A virtualization platform is required because OmniOS has limited support for modern consumer hardware. Commonly used virtualization tools include VMware Workstation and Oracle VM VirtualBox. These tools provide virtual hardware that is fully compatible with OmniOS and simplify installation and testing.

The system must run on a supported host operating system, such as Windows, Linux, or macOS, which will host the virtual machine. The host system's BIOS or UEFI firmware must have virtualization features enabled (Intel VT-x or AMD-V), and Secure Boot should be disabled to allow the OmniOS installer to boot correctly.

OmniOS uses the Image Packaging System (IPS) for software installation and updates. A terminal or console environment is required for system administration tasks, as OmniOS is primarily command-line based. For remote system management, SSH is recommended, although it is optional.

Finally, appropriate network configuration software is required within the virtual machine, such as NAT or bridged networking, to allow internet access for package installation and system updates.

Installation steps

Installation welcome to Oracle virtual box

sFirge 1 : welcome screen to the Oracle virtual box

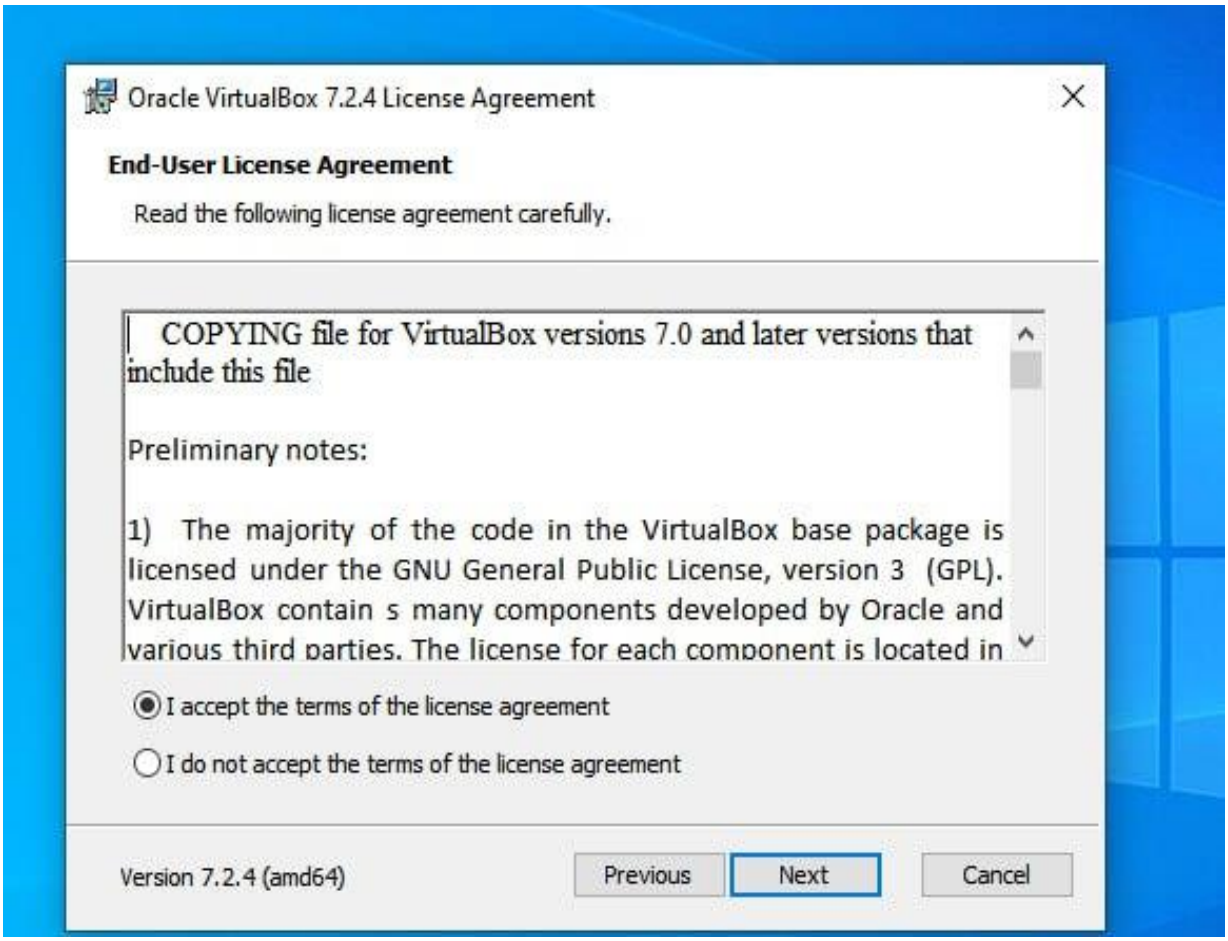


Figure 2 : accept the terms of the licence agreement screen

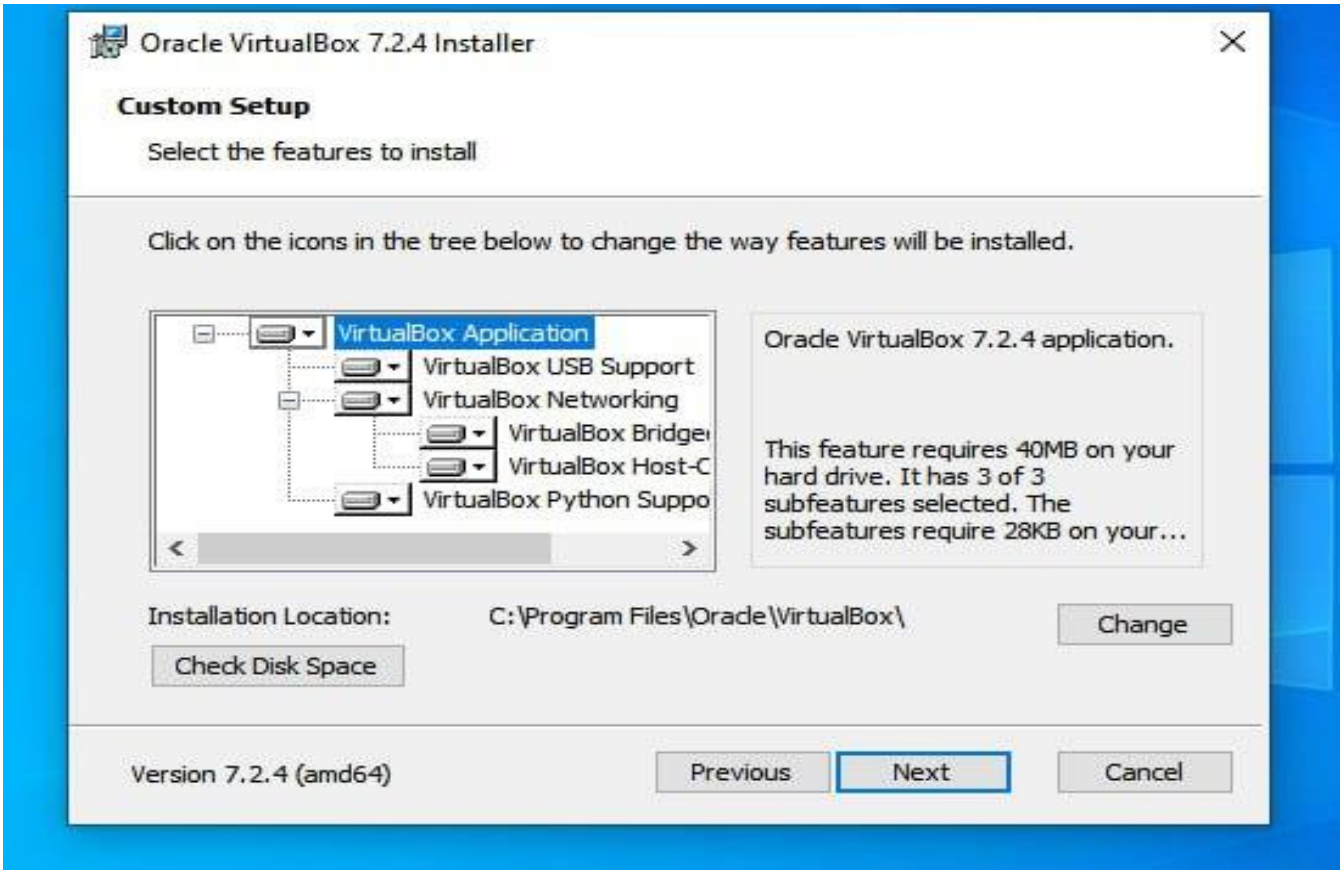


Fig 3: custom set up screen virtual box applications

Fig 4: warning screen networking disconnections

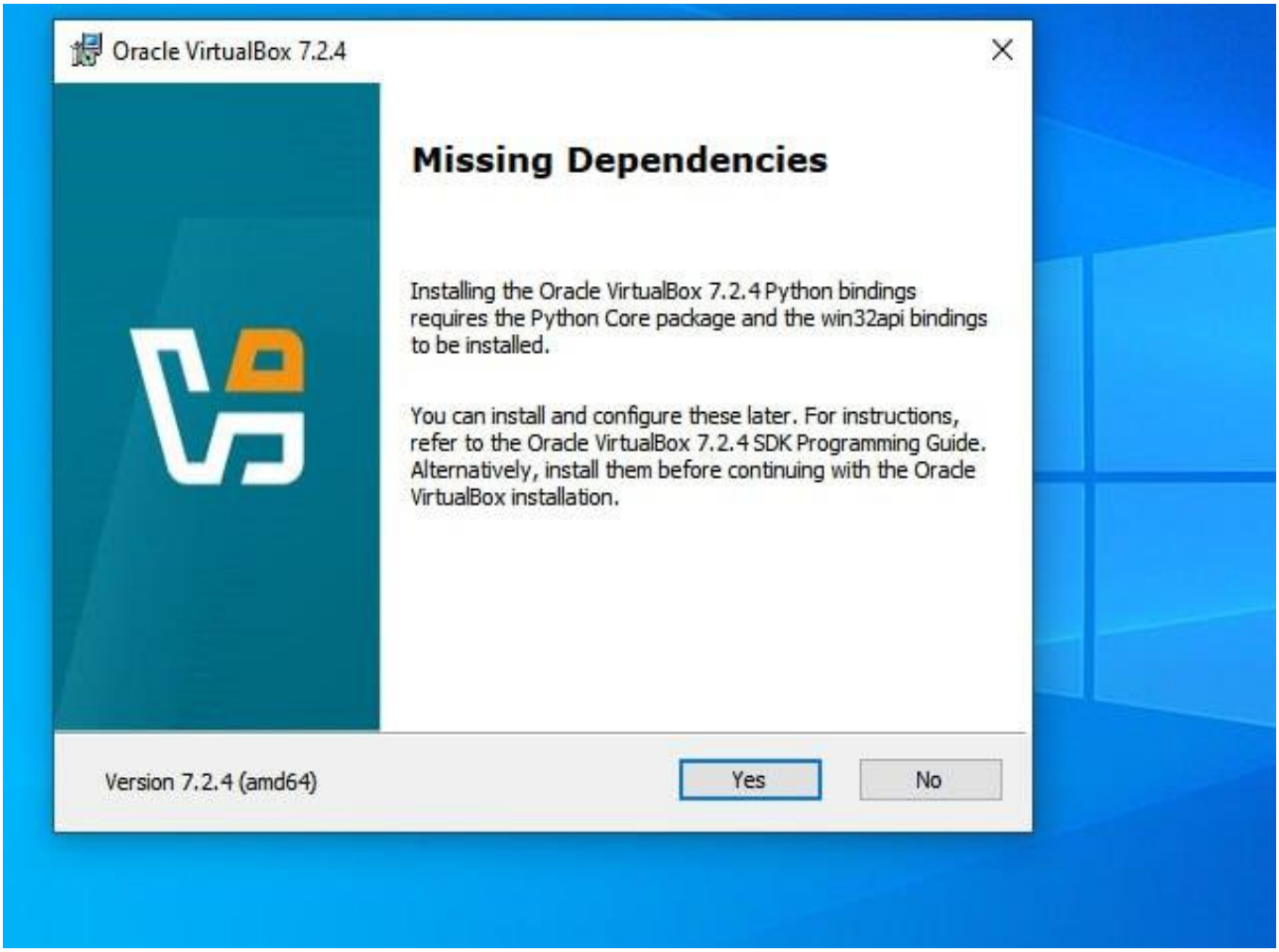


Fig 5: installing screen missing dependencies

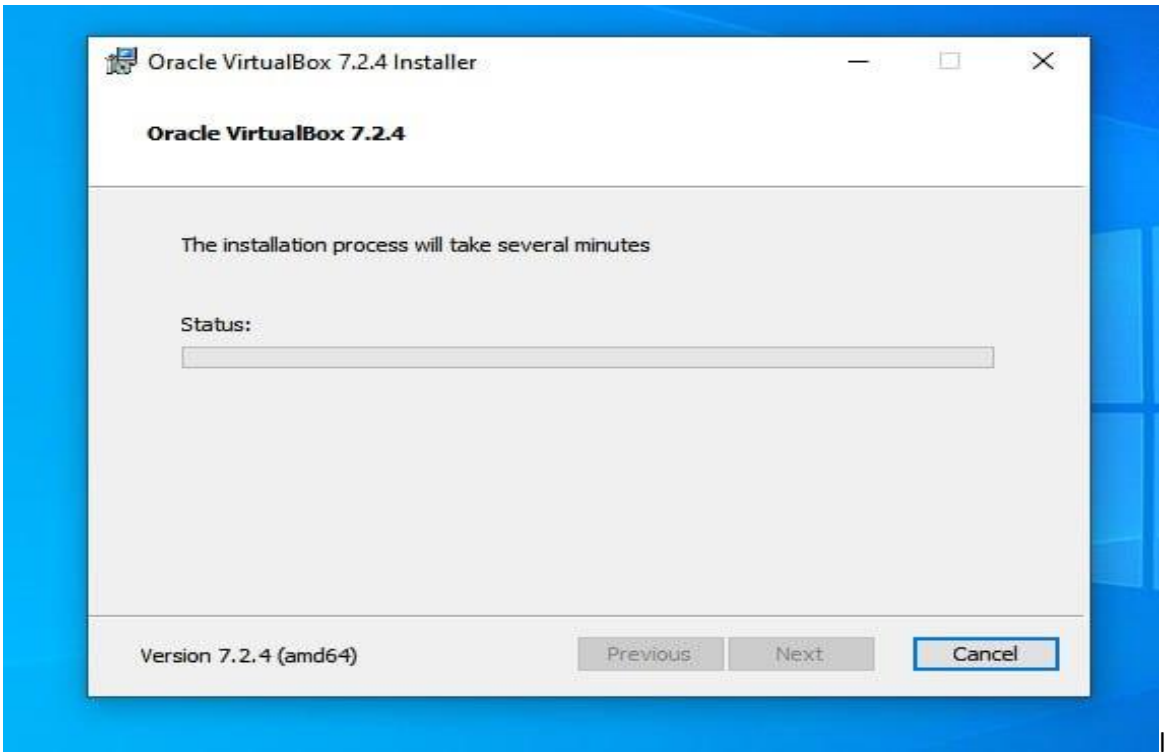


Fig 6:installation process screen Oracle virtual box Installation

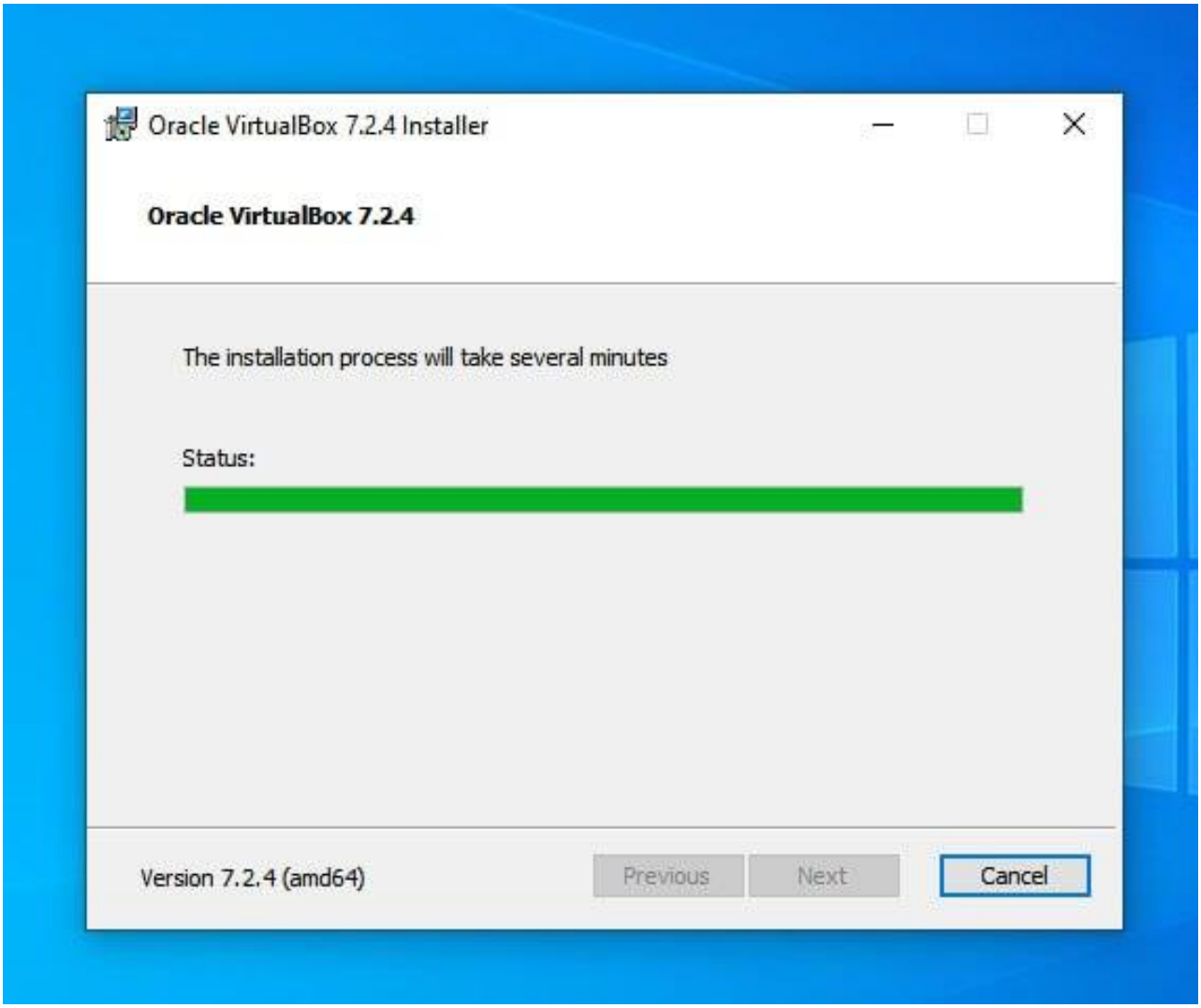


Fig 7: instaler screen Oracle virtual box Installation Process ended

Fig 8: instaler screen Oracle virtual box Installation been installed

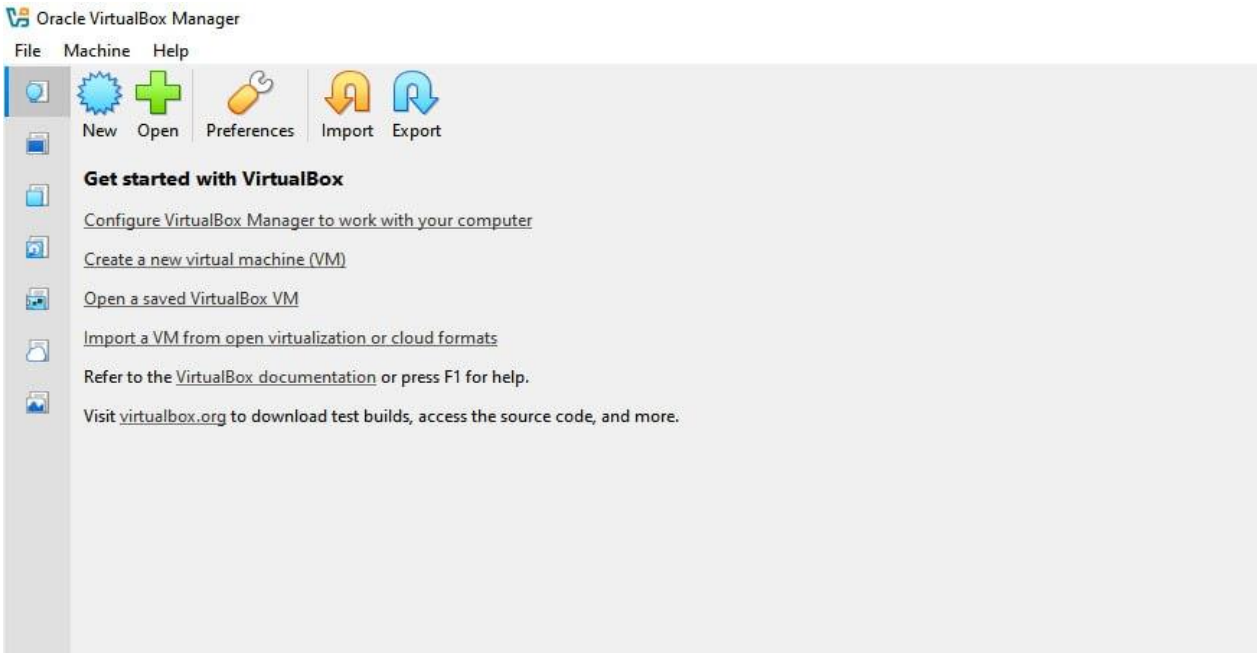


Fig 8: instaler screen Oracle virtual box Installation create new virtual VM

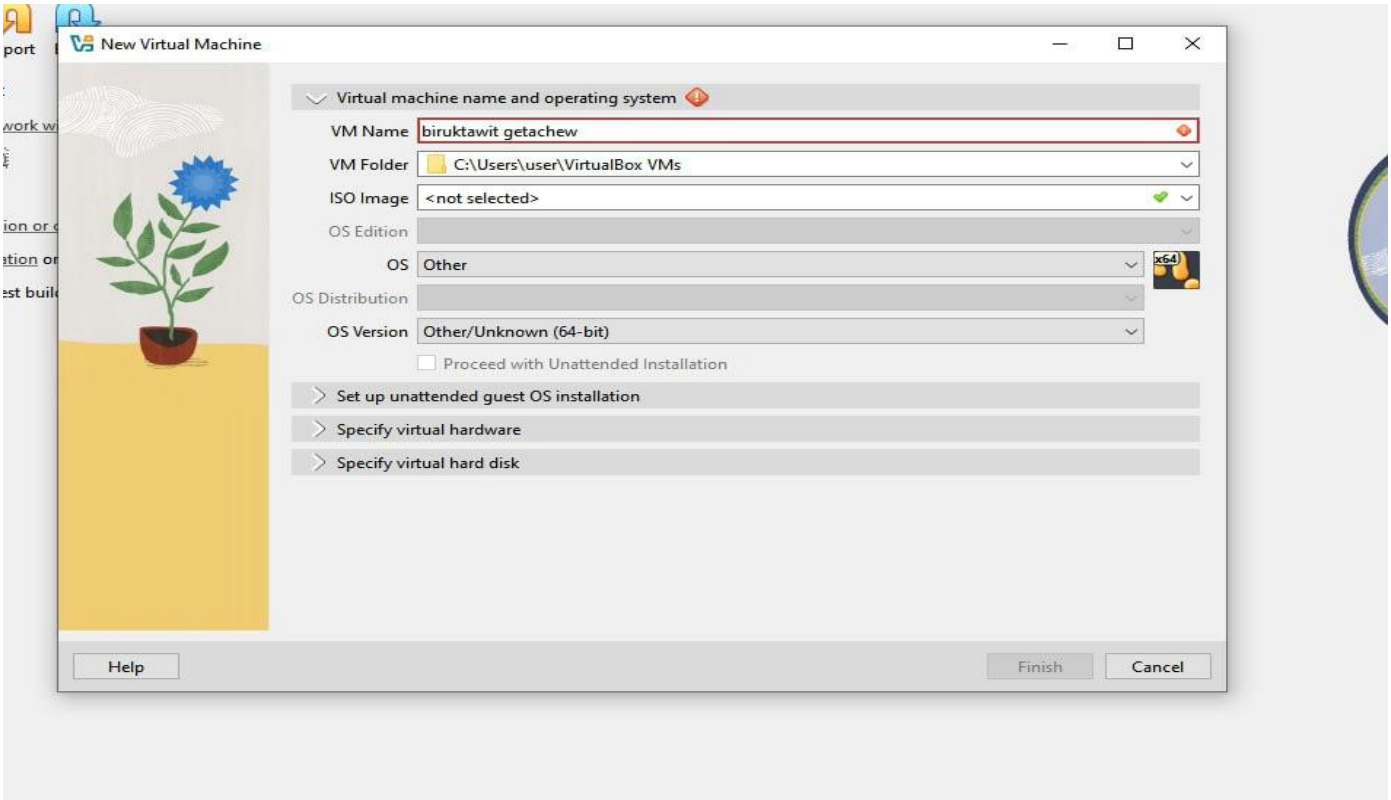


Fig 9 : Configuring VM hardware: allocating memory/CPU and setting disk size

2 Confirming Storage Attachment

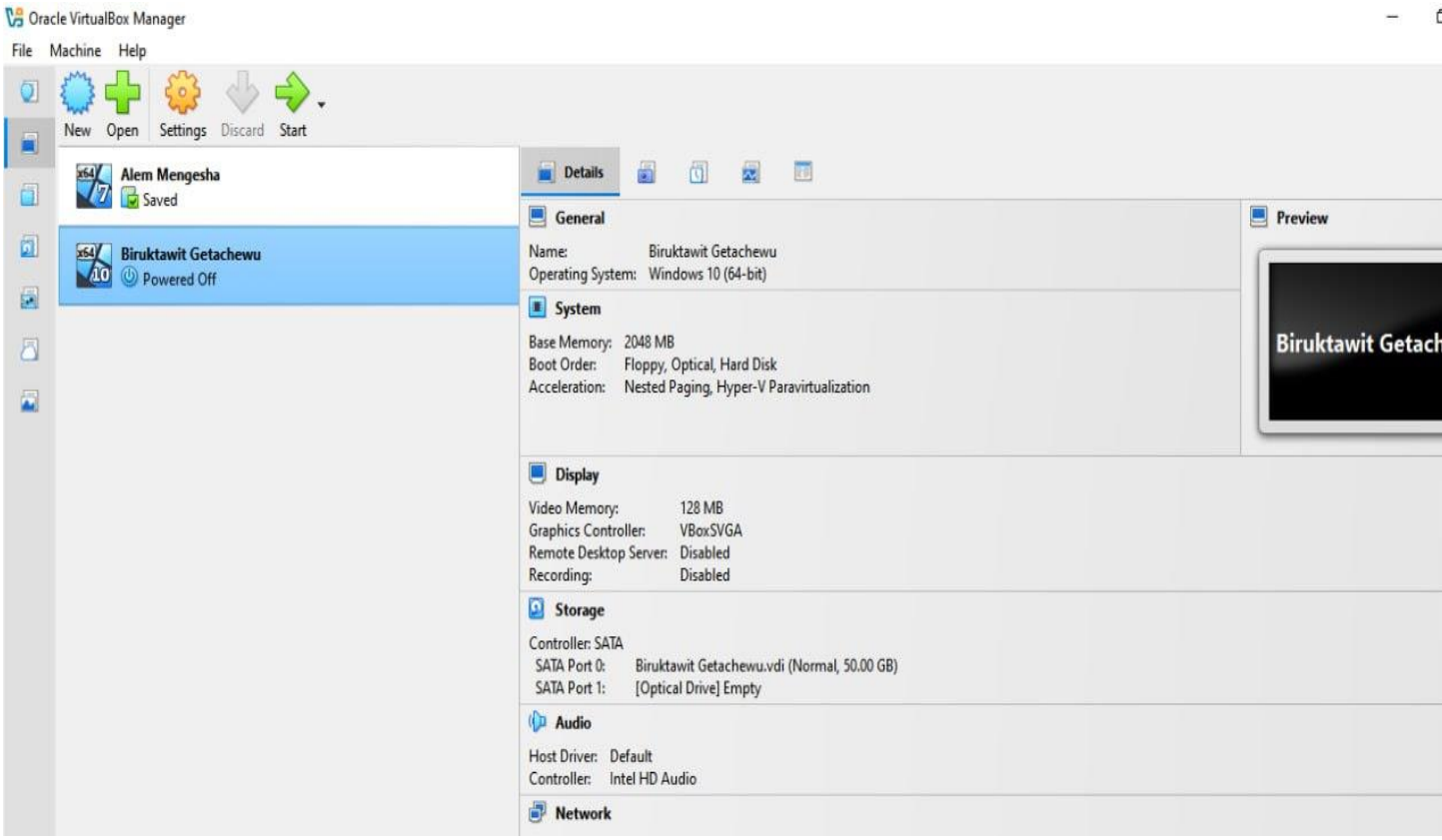


Fig 10:: instaler screen Oracle virtual box Installation been installed

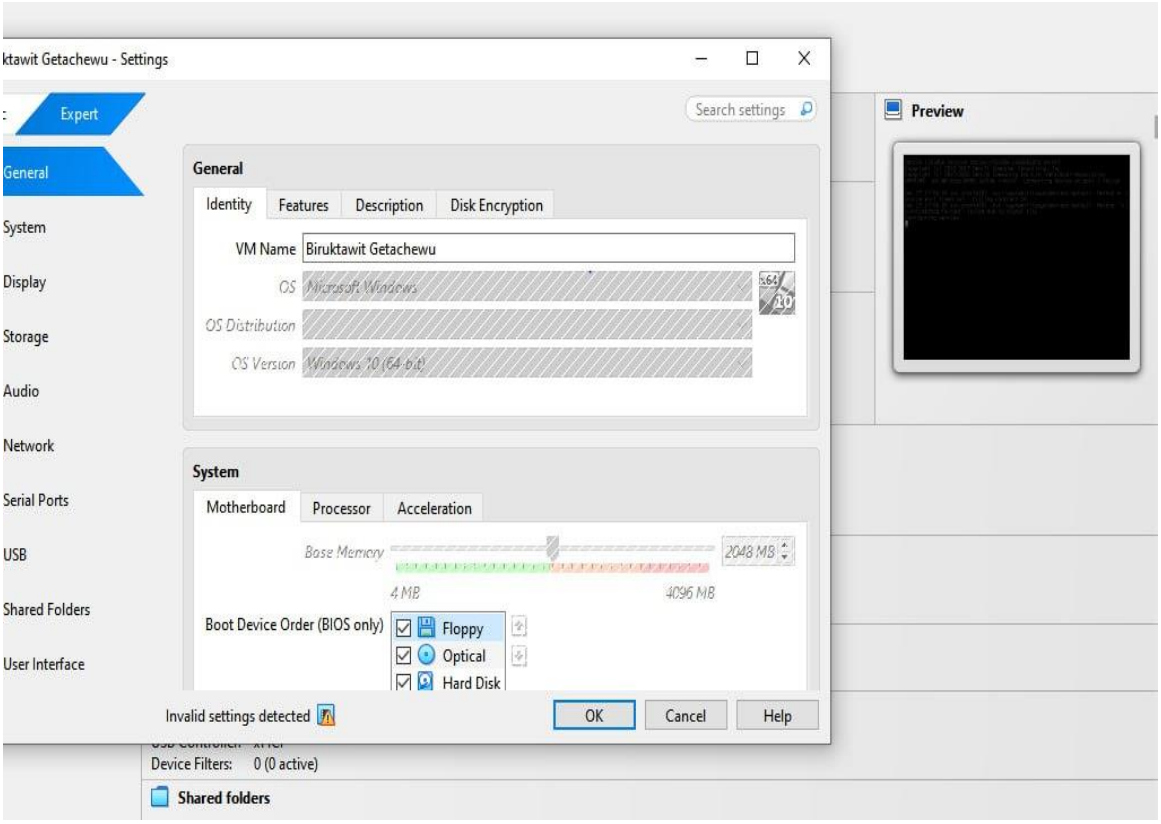


Fig 11

: instaler screen Oracle virtual box genenal

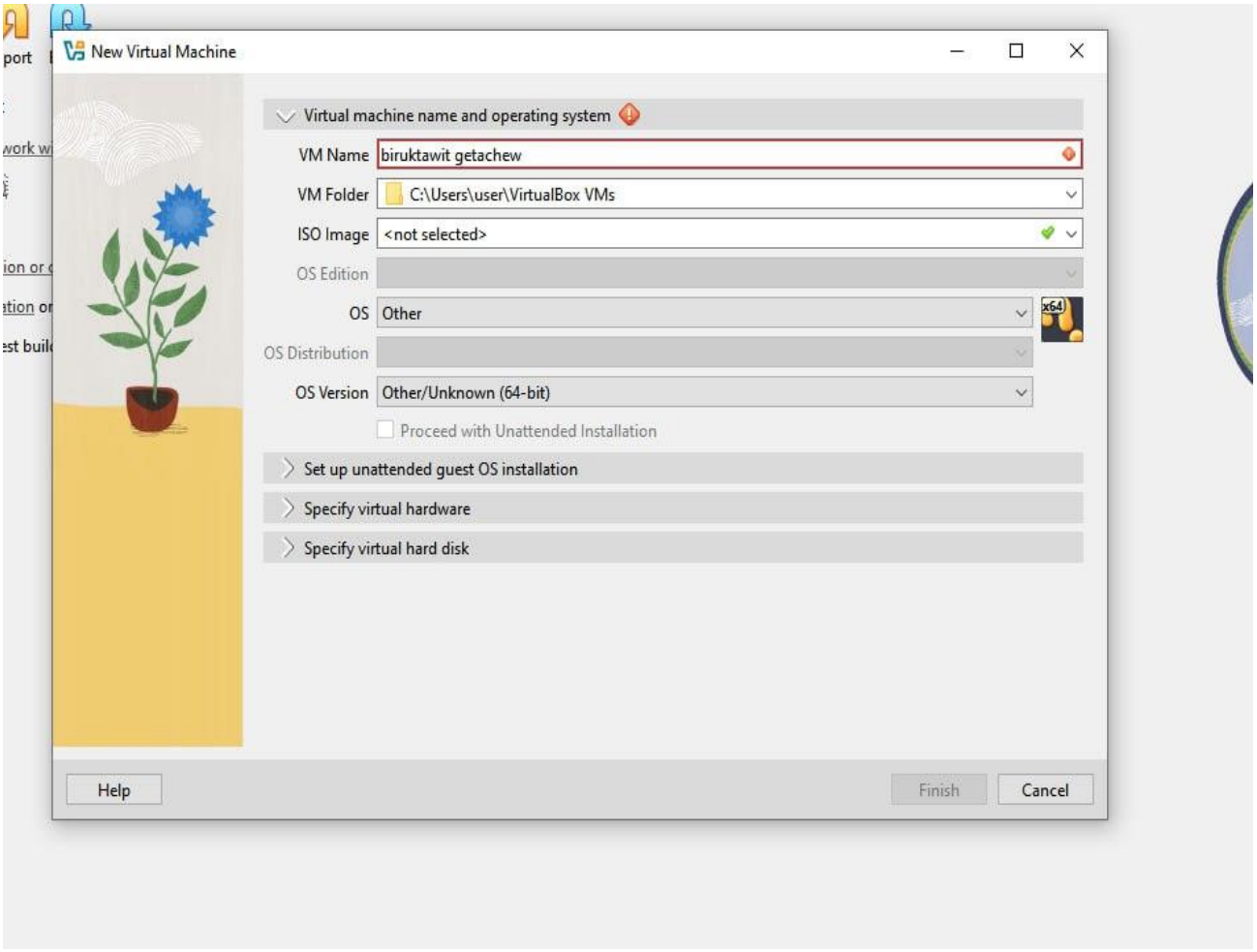


Fig 12: instaler screen Oracle virtual box Installation be

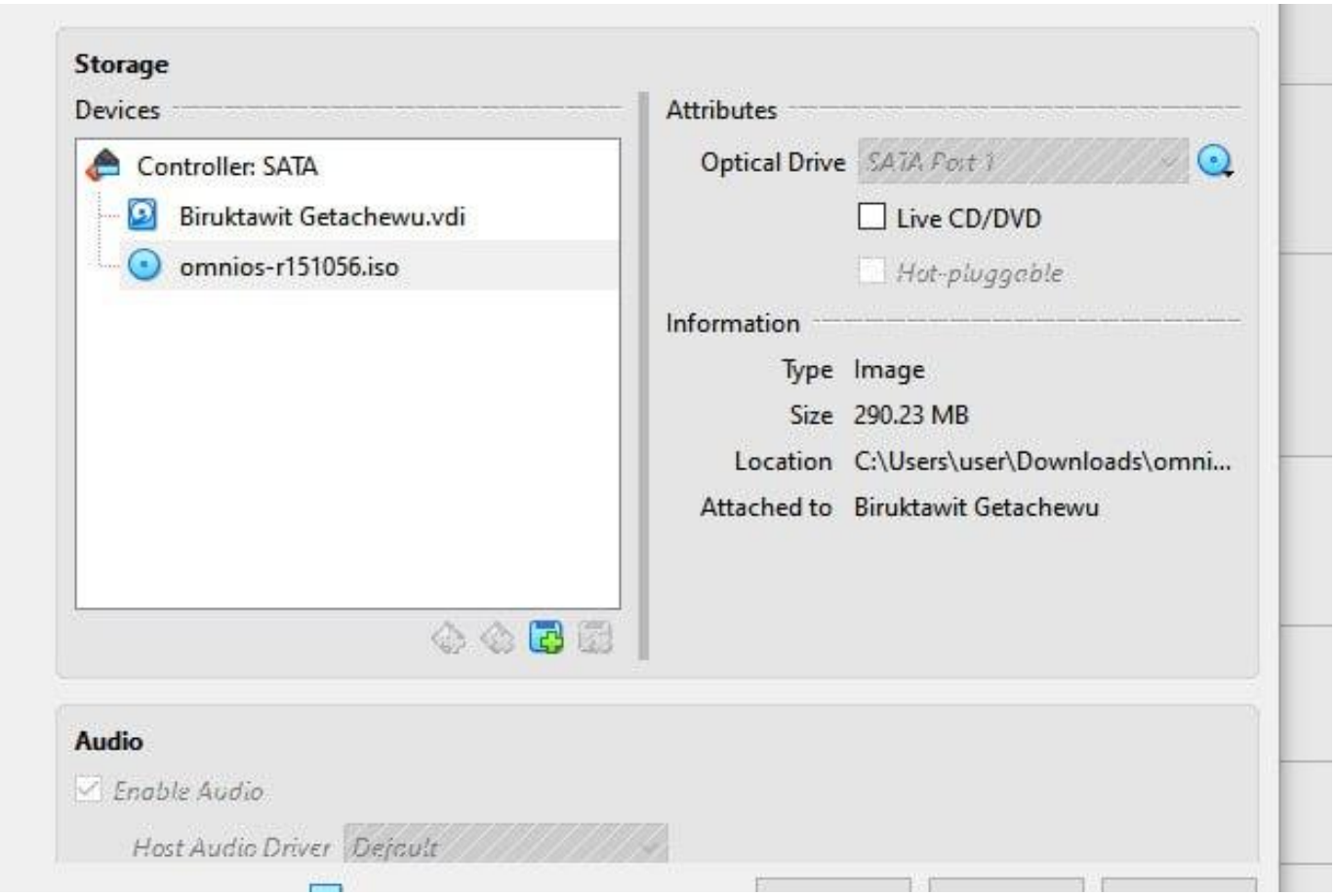
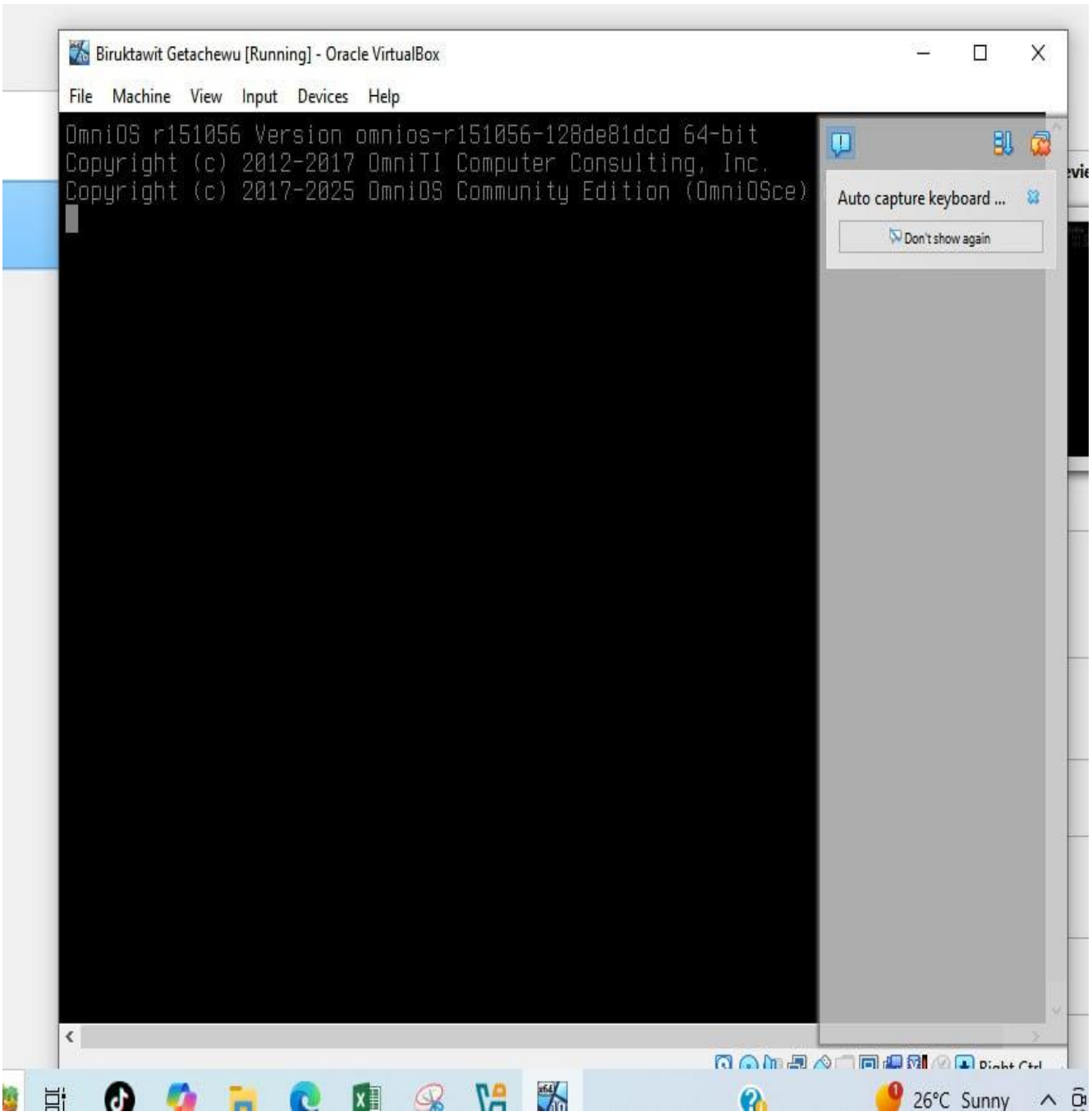
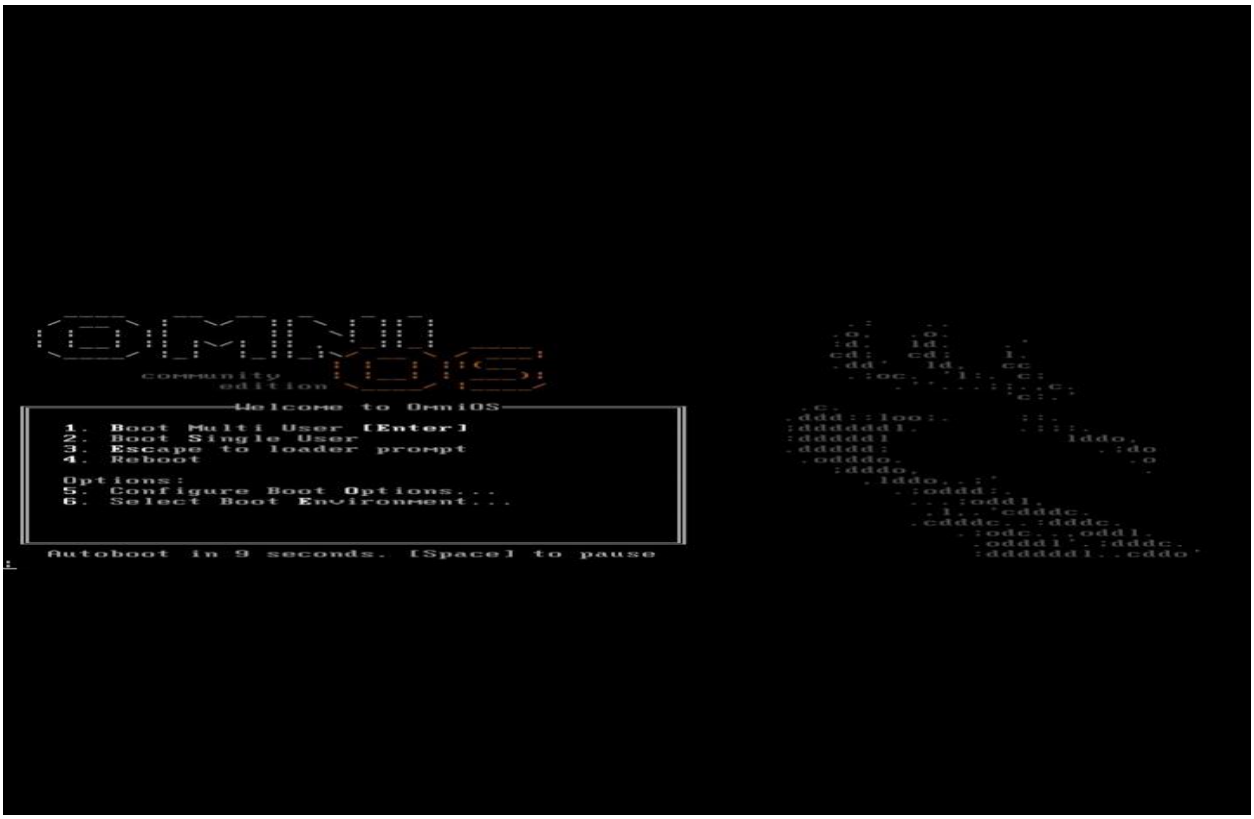


Fig 13: **storage attached document to OmniOS ISO file**


Biruktawit Getachewu [Running] - Oracle VirtualBox
File Machine View Input Devices Help
OmniOS r151056 Version omnios-r151056-128de81dcd 64-bit
Copyright (c) 2012-2017 OmniTI Computer Consulting, Inc.
Copyright (c) 2017-2025 OmniOS Community Edition (OmniOSce)
Auto capture keyboard ...
Don't show again
26°C Sunny

1) Boot the InstallerInsert your USB or CD into the machine.Boot and choose the device fromBIOS/UEFI boot menu.



❓ Problem Faced

Symptom: System installed OmniOS but now cannot boot.Observation: You mention processor is 3.2 GHz, and you “need 3.9 GHz”.

Goal: Explain clearly why boot fails and why processor speed may matter.

❓ Step-by-Step Reasoning of the Problem

1. CPU & Performance Misunderstanding

You said 3.2 GHz CPU, “need 3.9 GHz”.

Reality: OS boot does not literally require a CPU to be 3.9 GHz.

What matters are CPU features, not just clockspeed:

64-bit support (x86_64)SSE, SSE2, SSE3 instructionsNX/PAE supportAVX (on some modern kernels)

If the kernel was compiled for CPUs with higher instruction sets, an older CPU may fail to execute instructions → kernel panic → cannot boot.

So, 3.2 GHz vs 3.9 GHz is not about speed; it's about CPU generation and supported instructions.

Exam Point: "Boot failure can occur if CPU lacks required instructions, not just due to clock speed difference."

2. Bootloader Issue

OmniOS uses GRUB/bootadm as bootloader.

Possible issues:

Bootloader not installed on correct disk

Disk partitions mismatch (GPT vs MBR)

BIOS/UEFI boot mismatch

Effect: CPU executes bootloader, bootloader fails → cannot find kernel → system does not boot.
"Bootloader is a critical first step; if it fails, OS never starts, independent of CPU speed."

3. BIOS / UEFI Settings

Many OmniOS boot issues happen because BIOS/UEFI is misconfigured:

Installed OS in UEFI, but system is set to Legacy boot → cannot boot.

Installed OS in Legacy, but BIOS is UEFI → cannot boot
Disk Type Matters:

GPT → UEFI mode

MBR → Legacy BIOS mode

“Mismatch between installation mode and BIOS mode is a common cause of boot failure.”

4. Disk & Hardware Drivers

OmniOS may fail if:Disk controll(SATA/NVMe/RAID) unsupportedDisk is not detected properly at boot

Kernel cannot mount root filesystem → boot fails

CPU clock (3.2→3.9 GHz) is irrelevant herehardware support is the

□ Solutions for “Cannot Boot” Problem

1. Verify BIOS/UEFI Settings

Problem: Boot mode mismatch (UEFI vs Legacy)

Solution:Enter BIOS/UEFI setup (usually press F2, DEL, or F12 during boot).Check Boot Mode:

If OmniOS installed in UEFI mode, enable UEFI.

If installed in Legacy mode, enable CSM/Legacy.

Ensure the boot disk is listed first in boot priority.

□ Most common fix for “cannot boot” errors.

2. Check Disk Partitioning

Problem: Disk type mismatch with boot mode

Solution:

GPT disk → use UEFI bootMBR disk → use Legacy boot

If mismatch, reinstall or convert disk format (careful: data loss risk).

3. Reinstall / Repair Bootloader

Problem: Bootloader missing or corrupted

Solution:Boot OmniOS from USB Live / Rescue media.

Identify root disk:

Copy codeformatReinstall bootloader:Copy code

bootadm install-bootloader -v /dev/dsk/cXtXdX

Replace cXtXdX with your disk device.

Reboot and check if system boots.

4. Check CPU Compatibility

Problem: Older CPU may lack instructions required by kernelSolution:Verify CPU model:

Copy code prtdiag

Compare with OmniOS required CPU instructions (x86_64, SSE2+).

If missing, use:Compatible OmniOS build (older release)

Or a CPU with required features (modern 64-bit CPU)

Note: The CPU clock difference (3.2 GHz → 3.9 GHz) is not the cause; it's instructions support that matters.

Let me explain OmniOS primarily uses ZFS as its native filesystem. ZFS is fully supported because it provides advanced features such as data integrity through checksums, snapshots for backups, RAID-Z for storage

redundancy, compression, and deduplication. It is reliable and scalable, which makes it ideal for the system root and important data. All system installations on OmniOS are done on ZFS by default because it integrates directly with the kernel and ensures stability.

For legacy support, OmniOS can also use UFS (Unix File System). UFS is compatible with Solaris-based systems but lacks modern features such as snapshots or checksums, so it is rarely used for new installations.

OmniOS can access FAT32 drives easily, which is useful for USB sticks or external drives, though FAT32 has a maximum file size of 4 GB. NTFS drives are partially supported, typically read-only by default, but can be accessed fully if a third-party driver such as ntfs-3g is installed. exFAT is not natively supported and requires additional software to read or write.

Linux filesystems such as ext4 have limited support, often only through read-only access using special drivers. Btrfs, a modern Linux filesystem, is not supported at all. For Apple filesystems, HFS+ can be accessed with some limitations, typically read-only,

while APFS, the newer Apple filesystem, is completely unsupported on OmniOS.

In summary, ZFS is the default and recommended filesystem for system installation due to its reliability and advanced features. Other filesystems such as FAT32 or NTFS are supported mainly for external drives, while UFS, ext4, HFS+, exFAT, and APFS have either limited or no support and are not suitable for root/system

Advantages of Filesystems in OmniOS

ZFS – Provides excellent data integrity, supports snapshots for backups, handles very large storage pools, offers RAID-Z redundancy, and includes compression and deduplication. Fully integrated with OmniOS, making it ideal for system/root installation.

UFS – Stable, lightweight, and simple; works well on older hardware.

FAT32 – Widely supported across Windows, Mac, and Linux; simple and easy to use for external drives.

NTFS – Supports large files and Windows permissions; useful for sharing data with Windows systems.

exFAT – Supports large files and cross-platform use; suitable for USB/external drives.

ext4 – Robust, efficient, and supports large volumes (for Linux data).

HFS+ – Can be read for legacy Mac drives.

Disadvantages of Filesystems in OmniOS

ZFS – Requires more memory, may be complex for beginners, can be slower on older hardware if using features like compression or RAID-Z extensively.

UFS – Lacks modern features such as snapshots, checksums, compression, and deduplication; less reliable than ZFS.

FAT32 – Maximum file size is 4 GB; no permissions, journaling, or data integrity features.

NTFS – Native OmniOS support is limited; may be read-only without additional drivers; not suitable for system/root installation.

exFAT – Not natively supported in OmniOS; requires extra software; lacks journaling or snapshots.

ext4 – Limited support in OmniOS, often read-only; cannot be used for system installation.

Btrfs – Not supported at all in OmniOS.

HFS+ / APFS – HFS+ is partially supported (usually read-only); APFS is not supported, so neither can be used for root/system installation.

Conclusion

OmniOS primarily relies on ZFS as its native filesystem because it provides high reliability, advanced data management features, and excellent protection against data corruption. Other filesystems such as UFS, FAT32, NTFS, and HFS+ can be used for specific external or legacy purposes, but they are not suitable for system installation. ZFS ensures that the system is robust, scalable, and able to handle large storage pools efficiently, making it the preferred choice for both servers and enterprise environments.

Future Outlook / Recommendation

For future deployments, it is recommended to continue using ZFS as the primary filesystem due to its flexibility, snapshot capabilities, and fault tolerance. For external

storage or cross-platform sharing, FAT32 or NTFS may be used, but with awareness of their limitations. As storage technology evolves, ZFS will continue to provide strong support for modern enterprise needs, including cloud and virtualization environments. Newer filesystems like APFS or Btrfs could be considered only if OmniOS or compatible drivers provide full support, but currently, ZFS remains the most reliable and feature-complete option.

What is Virtualization?

Virtualization is a technology that allows a single physical computer (host) to run multiple virtual machines (VMs), each with its own operating system and applications, as if they were separate physical computers. It creates an abstract layer between hardware and software, so multiple OS environments can coexist on the same hardware.

Why Virtualization?

Resource Efficiency: Multiple VMs can share the same hardware, reducing the need for extraphysical machines.

Isolation: Each VM runs independently, so a crash or security issue in one VM does not affect others.

Flexibility & Testing: Allows testing of different operating systems, software, or configurations without affecting the host system.

Cost Savings: Reduces hardware and energy costs by consolidating workloads onto fewer physical servers.Scalability: Easy to deploy, move, or clone virtual machines for cloud computing or enterprise environments.

How Virtualization Works?

A hypervisor (software layer) manages the VMs and allocates hardware resources like CPU, memory, and storage.

There are two types of hypervisors:

Type 1 (Bare-metal): Runs directly on hardware (e.g., VMware ESXi, Hyper-V).

Type 2 (Hosted): Runs on top of a host OS (e.g., VirtualBox, VMware Workstation).

The hypervisor abstracts the physical hardware so each VM believes it has its own dedicated CPU, memory, disk, and network interfaces.

Modern OSes include virtualization support at both hardware level (Intel VT-x / AMD-V) and software level (hypervisor integration) to improve performance.