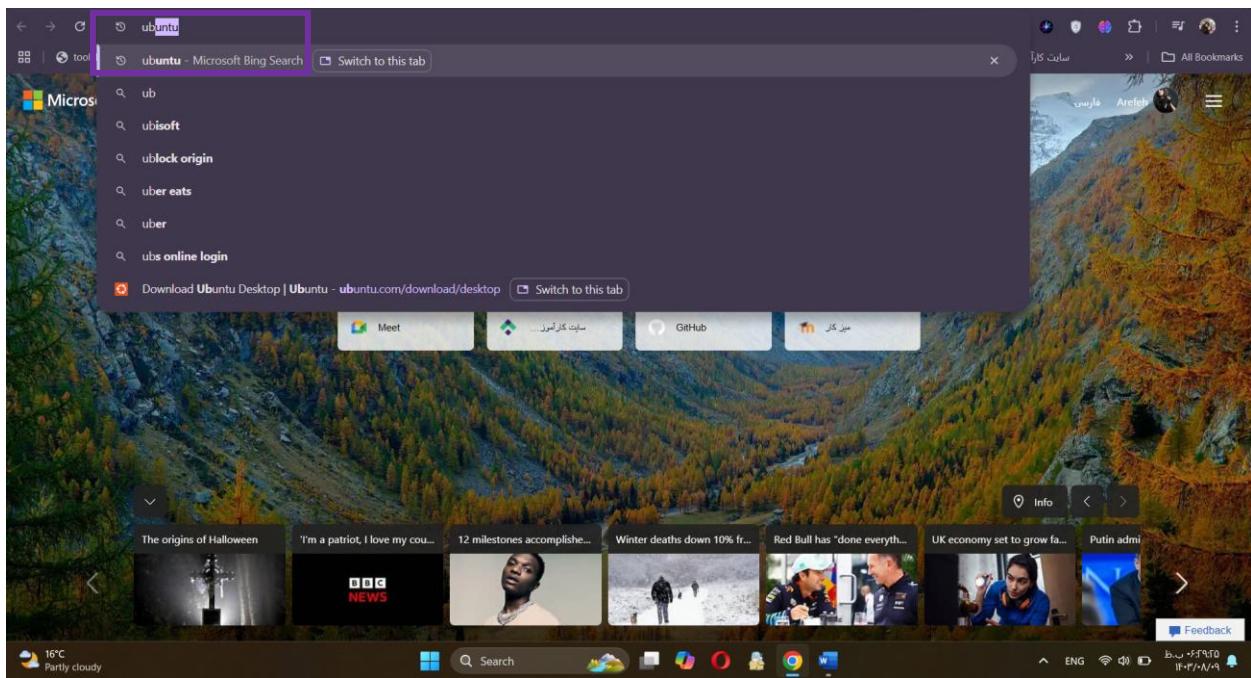


فهرست

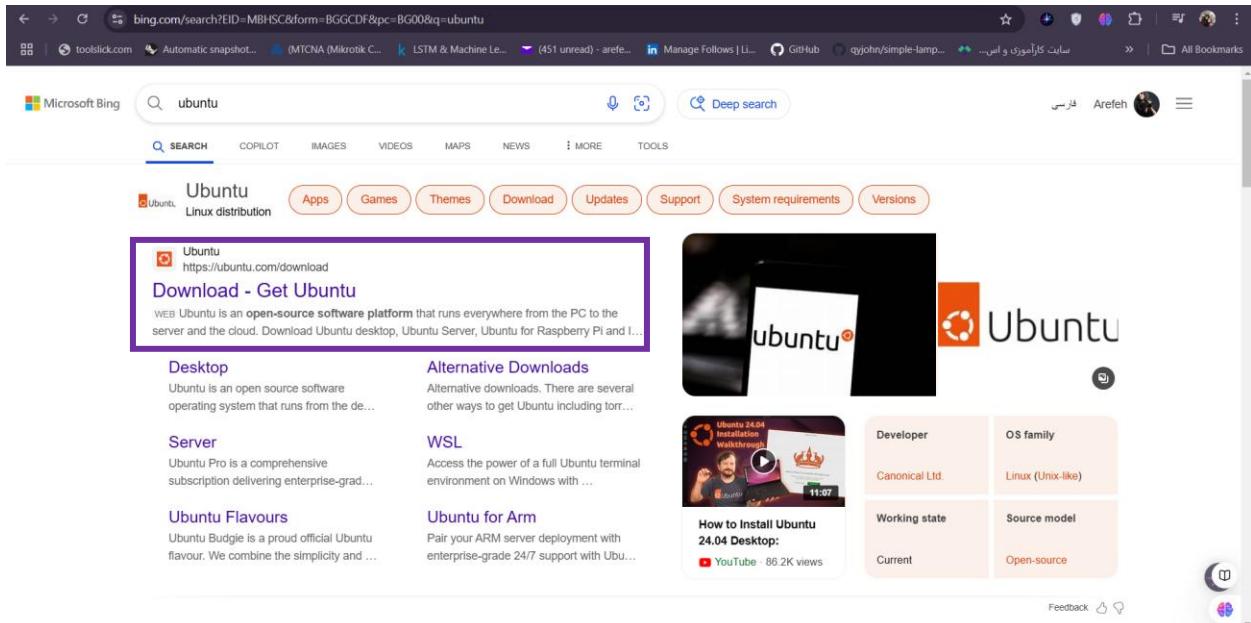
2.....	نصب لینوکس اوبونتو روی VMWare
6.....	نصب اوبونتو روی سیستم مجازی ساز(VMware)
15.....	Docker
17.....	نصب داکر روی لینوکس
22.....	نصب وبسرور، PHP و پایگاه داده (LAMP Stack)
25.....	نصب ابزارهای ویرایش کد
26.....	ایجاد پروژه کراد لیست برنامه روزانه
32.....	داکرایز کردن پروژه و دیتابیس پروژه با شبکه خارجی کانتینترها
37.....	تست دیتاها با POSTMAN
42.....	آپلود داکیومنت پروژه روی گیتهاب
51.....	References

نصب لینوکس اوبونتو روی VMWare

اول دانلود iso سیستم عامل اوبونتو از وبسایت رسمی ubuntu



The screenshot shows the Microsoft Edge browser interface. The address bar contains the search term "ubuntu". Below the address bar, a search card for "ubuntu" is displayed, showing a list of search results including "Ubuntu - Microsoft Bing Search", "Download Ubuntu Desktop | Ubuntu - ubuntu.com/download/desktop", and other links. The main content area of the browser shows a news feed with various headlines like "The origins of Halloween", "I'm a patriot, I love my cou...", "12 milestones accomplishe...", "Winter deaths down 10% fr...", "Red Bull has 'done everyth...', "UK economy set to grow fa...", and "Putin admin...". At the bottom of the browser window, there is a toolbar with icons for Meet, GitHub, and other Microsoft services, along with weather information (16°C Partly cloudy) and system status indicators.



The screenshot shows the Microsoft Bing search results page for the query "ubuntu". The search bar at the top contains "ubuntu". Below the search bar, there are tabs for "SEARCH", "COPilot", "IMAGES", "VIDEOS", "MAPS", "NEWS", "MORE", and "TOOLS". The main content area features a large image of the Ubuntu logo. Below the image, there is a section titled "Ubuntu" with a sub-section "Ubuntu Linux distribution". This section includes a link to "https://ubuntu.com/download" and a button labeled "Download - Get Ubuntu". To the right of this, there is a video thumbnail for "Ubuntu 24.04 LTS Release Washington" and a "Deep search" button. On the left, there are sections for "Desktop", "Server", and "Ubuntu Flavours". On the right, there are tables for "Developer" (Canonical Ltd.), "OS family" (Linux (Unix-like)), "Working state" (Current), and "Source model" (Open-source). The bottom of the page includes a "Feedback" button and social media sharing icons.

The screenshot shows the Ubuntu download page at ubuntu.com/download. The top navigation bar includes links for Canonical Ubuntu, Products, Use cases, Support, Community, Get Ubuntu (selected), All Canonical, Sign in, and a search bar. Below the navigation is a secondary menu with options for Downloads, Desktop, Server, Core, and Cloud. The main content area features a large heading "Download Ubuntu" and a sub-headline: "Ubuntu is the world's favourite Linux operating system. Run it on your laptop, workstation, server or IoT device, with five years of free security updates." To the right is a photograph of two laptops, one open and displaying the Ubuntu desktop environment.

Download Ubuntu

Ubuntu is the world's favourite Linux operating system. Run it on your laptop, workstation, server or IoT device, with five years of free security updates.



CHOOSE THE OS YOU NEED

Desktop >

This screenshot shows the "CHOOSE THE OS YOU NEED" section of the Ubuntu website. It lists several options: Desktop, Server, Raspberry Pi, Ubuntu for IoT, Develop on Ubuntu, and Windows & macOS. The "Desktop" option is highlighted with a green button labeled "Download Ubuntu Desktop". To the right, there is a sidebar for "Ubuntu-ready PC and laptops" and a "QUICK LINKS" section with links to upgrade guides, installation tutorials, and case studies. The URL in the address bar is <https://ubuntu.com/navigation#get-ubuntu-navigation>.

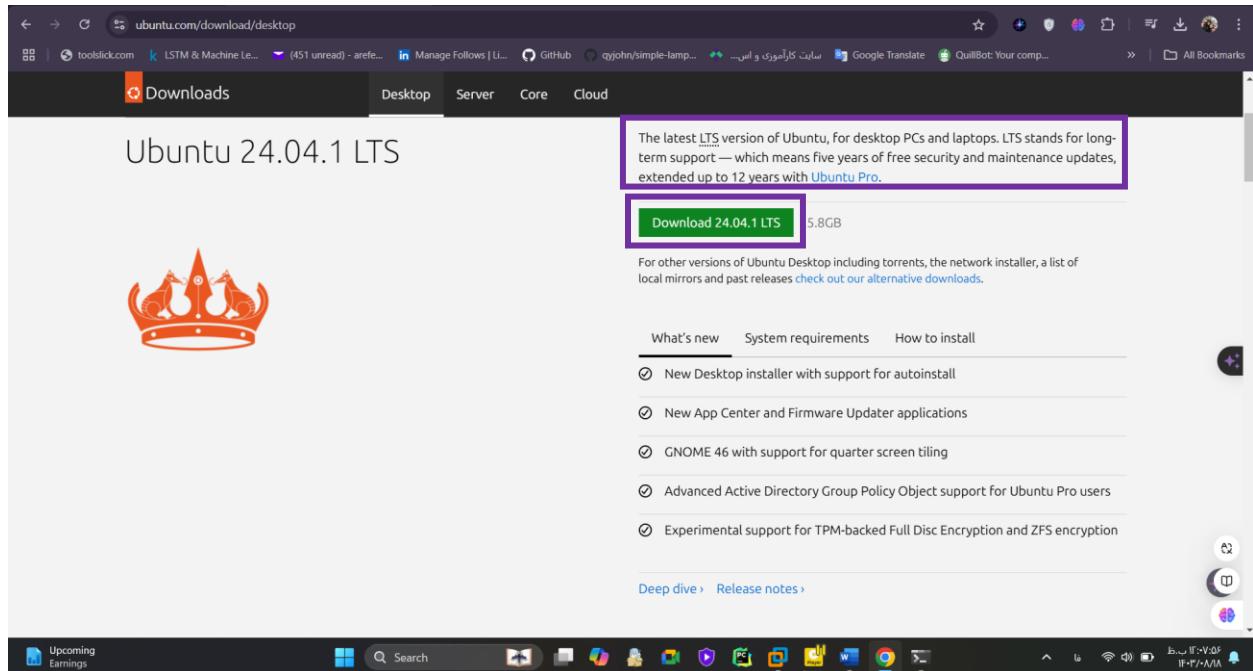
CHOOSE THE OS YOU NEED

Desktop >

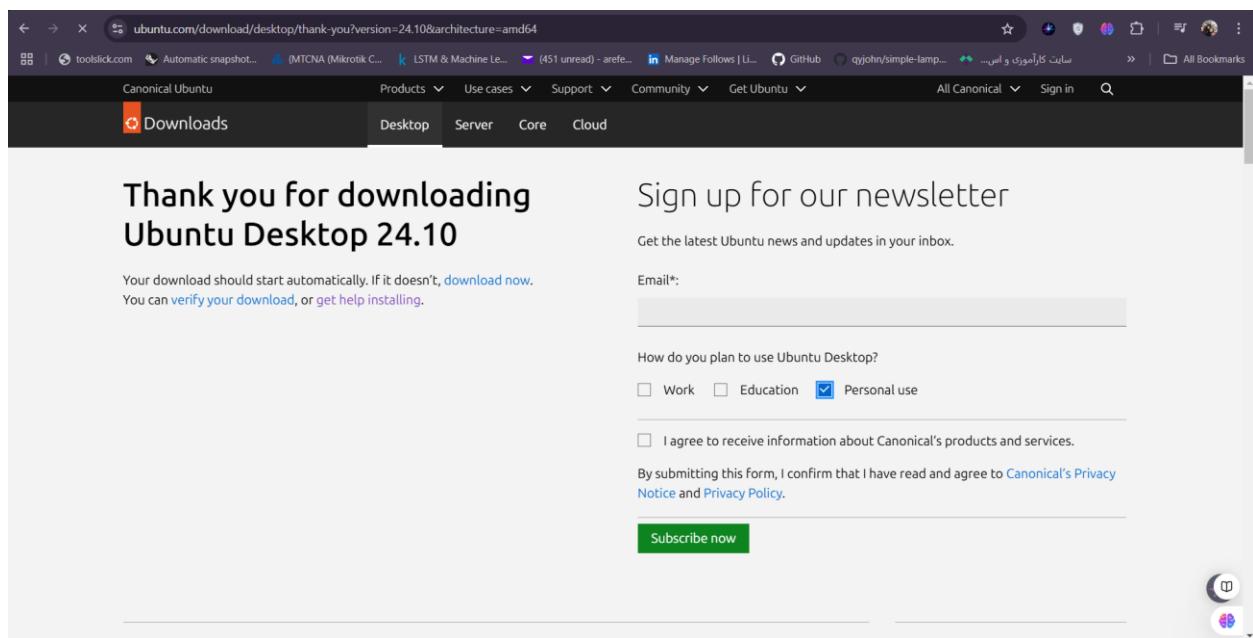
This screenshot shows the same "CHOOSE THE OS YOU NEED" section on a Windows desktop. The "Desktop" option is again highlighted with a green "Download Ubuntu Desktop" button. The desktop environment includes a taskbar with various icons and a weather widget showing "16°C Partly cloudy". The URL in the address bar is <https://ubuntu.com/navigation#get-ubuntu-navigation>.

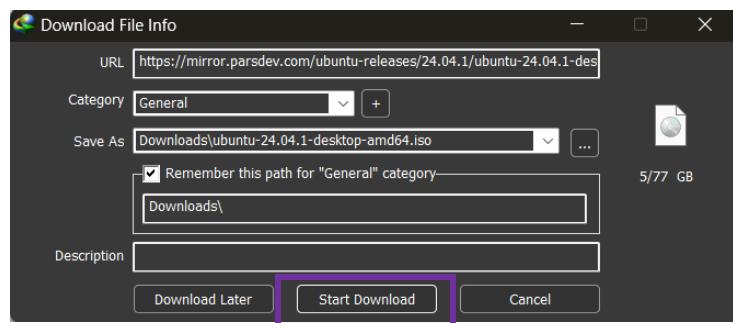
دانلود آخرین ورژن لینوکس اوبونتو

علت :LST



و بعد این صفحه شروع به دانلود فایل میشے.





باید کمی منتظر بمانیم.....

در windows terminal میشه دستورات لینوکسی رو اجرا کرد.

Ls/ dir

Clear/cls

```
PS C:\Users\arefe> ls

Directory: C:\Users\arefe

Mode                LastWriteTime         Length Name
----                -----        ----
d-----       6/20/2024   1:09 AM           .anaconda
d-----       10/2/2024  6:35 PM           .arduinoIDE
d-----       7/8/2024  12:58 AM           .astropy
d-----      10/27/2024  8:32 PM           .conda
```

```
C:\Users\arefe>clear
'clear' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\arefe>
```

دستورات لینوکسی رو میشه در ویندوز هم استفاده کرد

بهتره از لینوکس روی مجازی ساز پیش بریم

منطق لینوکس و کار با لینوکس ها

دستور و منطق عوض نمیشه ممکنه یکی دو تاش deprecate بشه

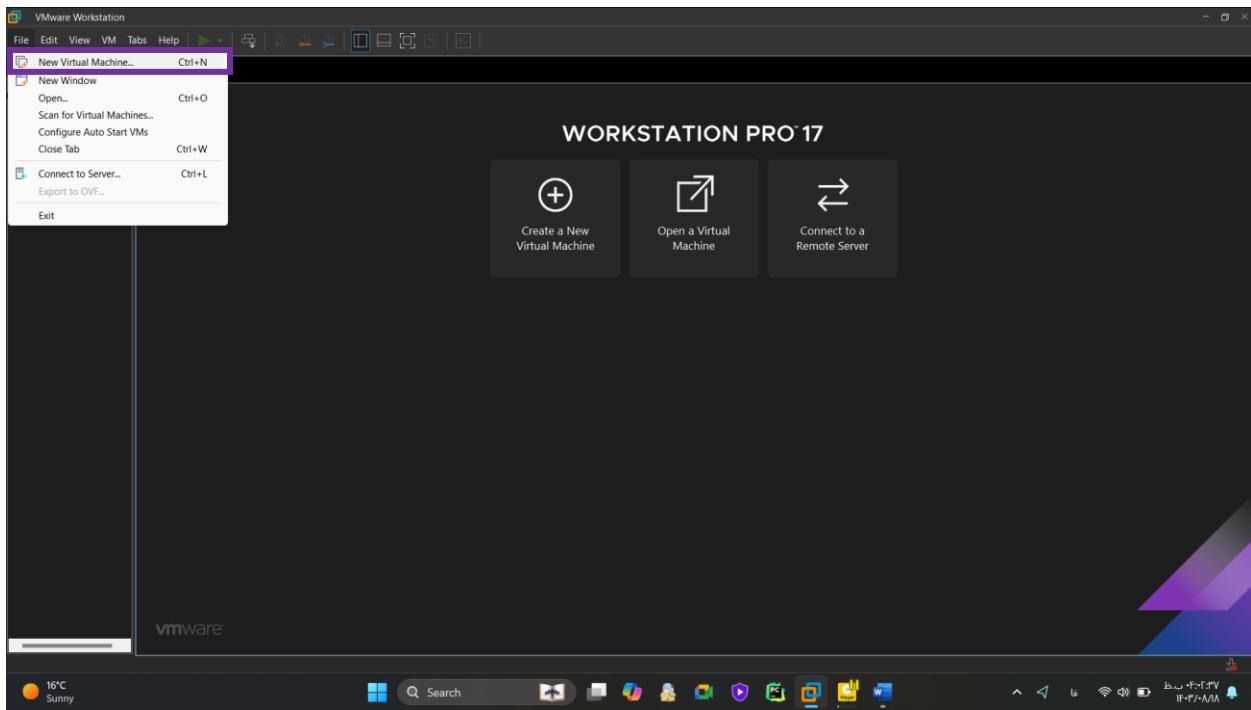
Mint

Debian

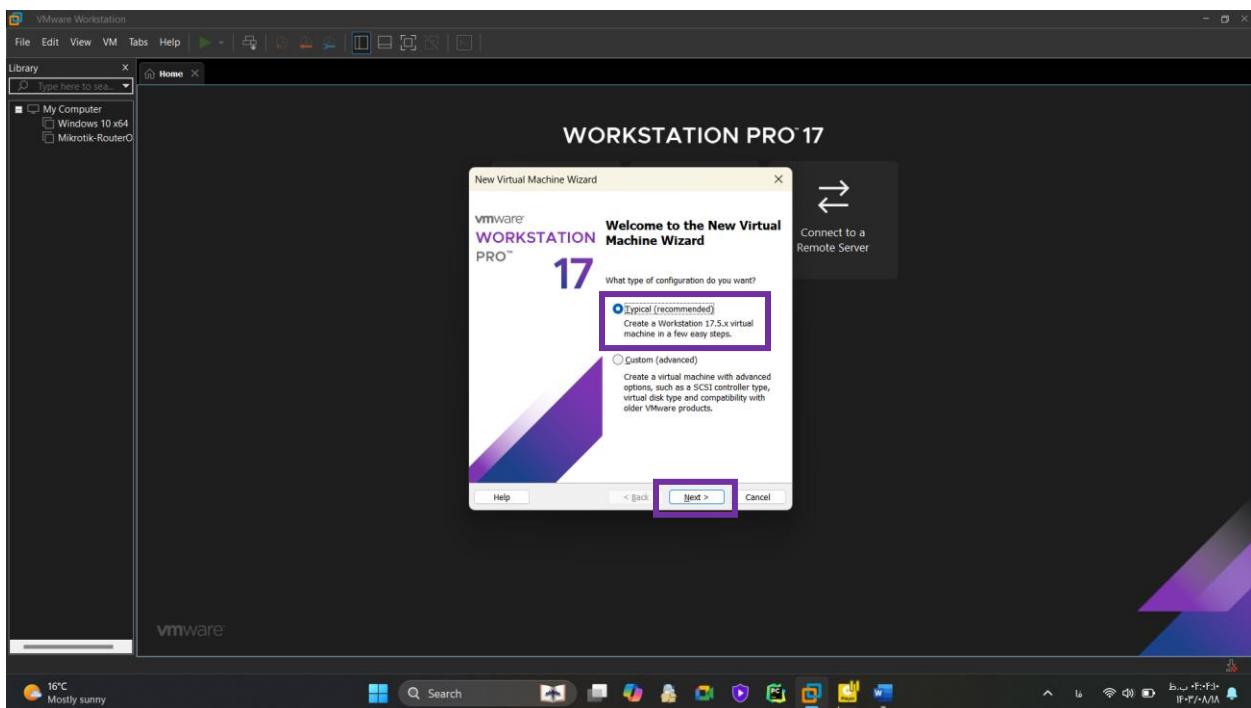
Kali

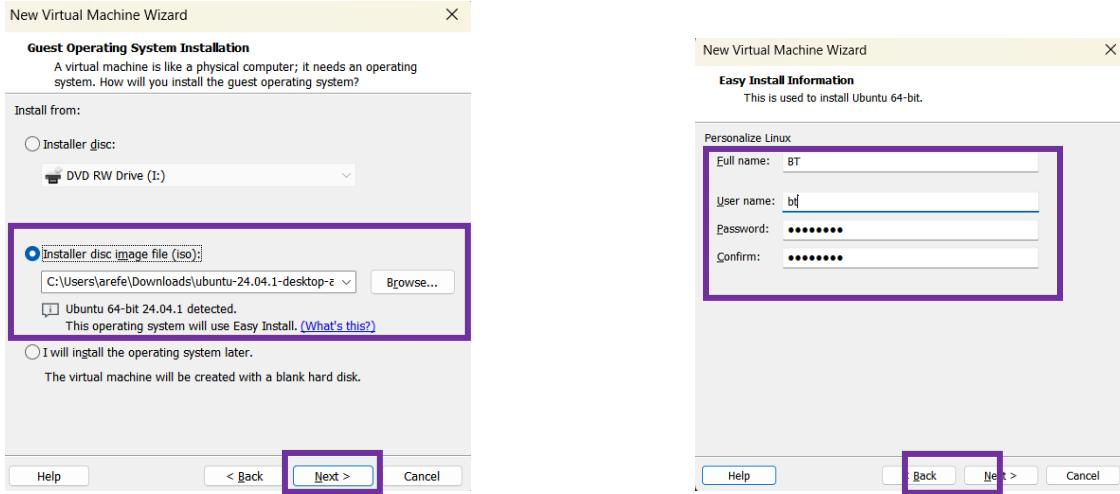
میشه در virtual box هم نصب کرد ولی خب ما VMware رو انتخاب کردیم

نصب اوبونتو روی سیستم مجازی ساز (VMware)

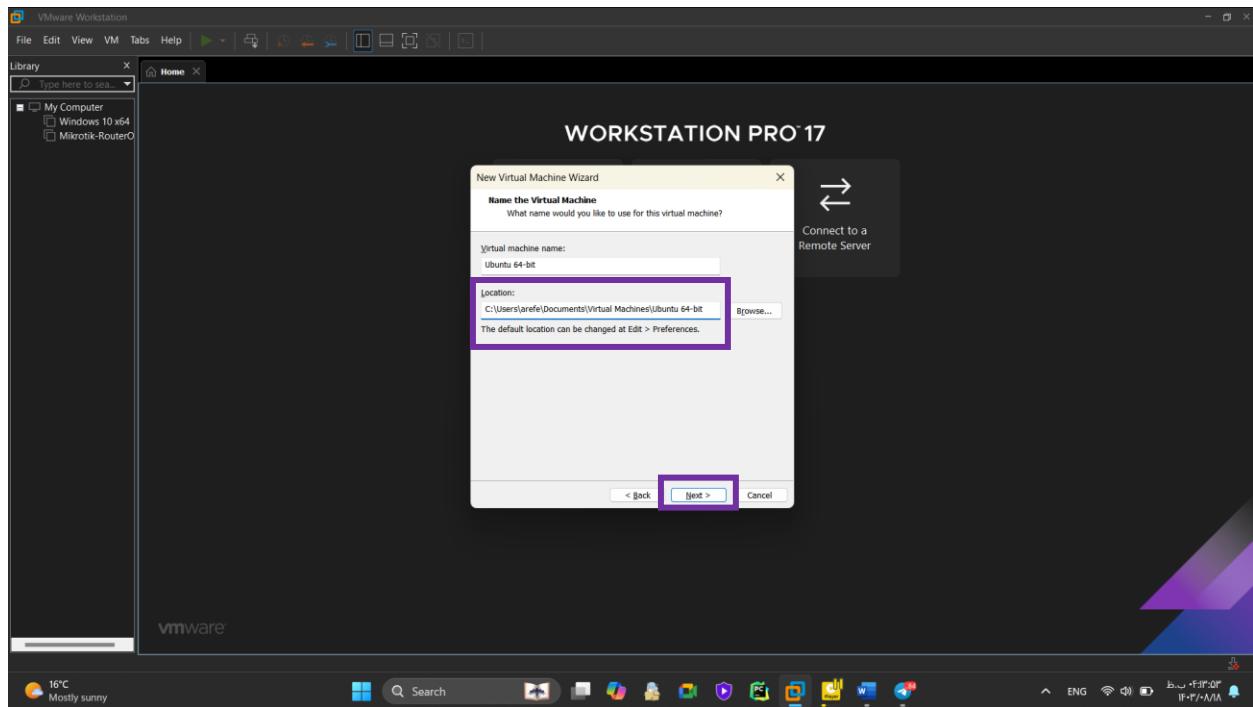


خودش جلو میره recommended هم هست.

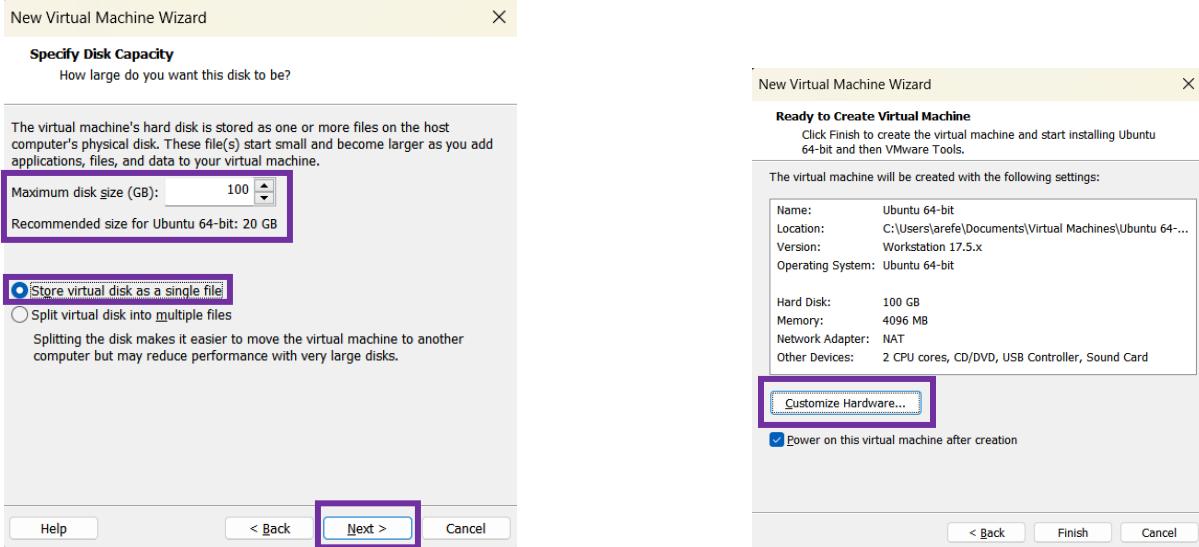




میشه اسم دلخواه هم برای ماشن مجازی مشخص کرد که ما با همین ادامه میدهیم. برای محل ذخیره سازی باید درایو مورد نظر حافظه ۸۰ گیگ خالی داشته باشه تا بعدا به مشکل برخوریم.



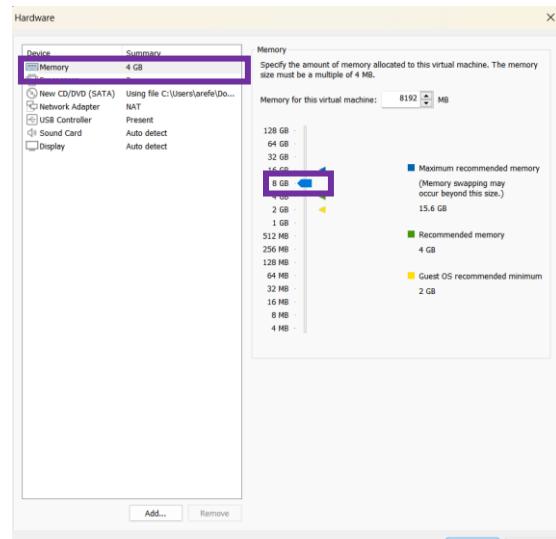
۱۰۰ گیگ در نظر میگریم و گزینه اول رو میزنیم که روی یک فایل بیشتر اویونتو رو ذخیره نکنه



روی customize Hardware... که کلیک کنید میشه تنظیمات سخت افزاری رو به صورت دلخواه تنظیم کنیم.

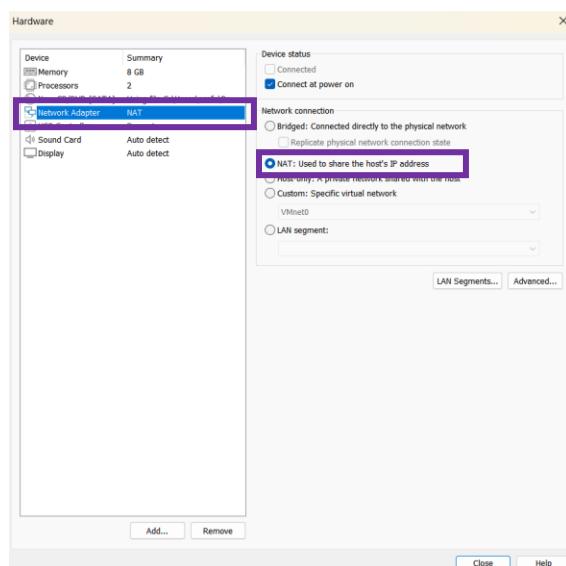
① Device specifications	
Device name	Trinity
Processor	AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx 2.30 GHz
Installed RAM	20/0 GB (17/9 GB usable)
Device ID	F58D08EC-8A62-441C-93A1-8B4BF39FC96D
Product ID	00331-10000-00001-AA820
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

مثال رم من ۲۰ هست میتونم تا ۱۰ گیگ رم بدم به ماشین مجازی م



نصب کردن نیست و NAT اینترنش رو از ویندوز میگیره خیلی مهمه. اگر NAT روی سیستم ویندوز گذر از تحریم فعال باشه روی این لینوکس هم فعال میشه احتیاج به

ولی در حالت bridged ماشین مجازی وصل میشه به مودم و مثله حالت قبلی نیست.



میشه 1Gig گرافیک براساس کارت گرافیک‌مون در نظر بگیریم چون تا 2Gig گرافیک داریم

The left window shows the following settings:

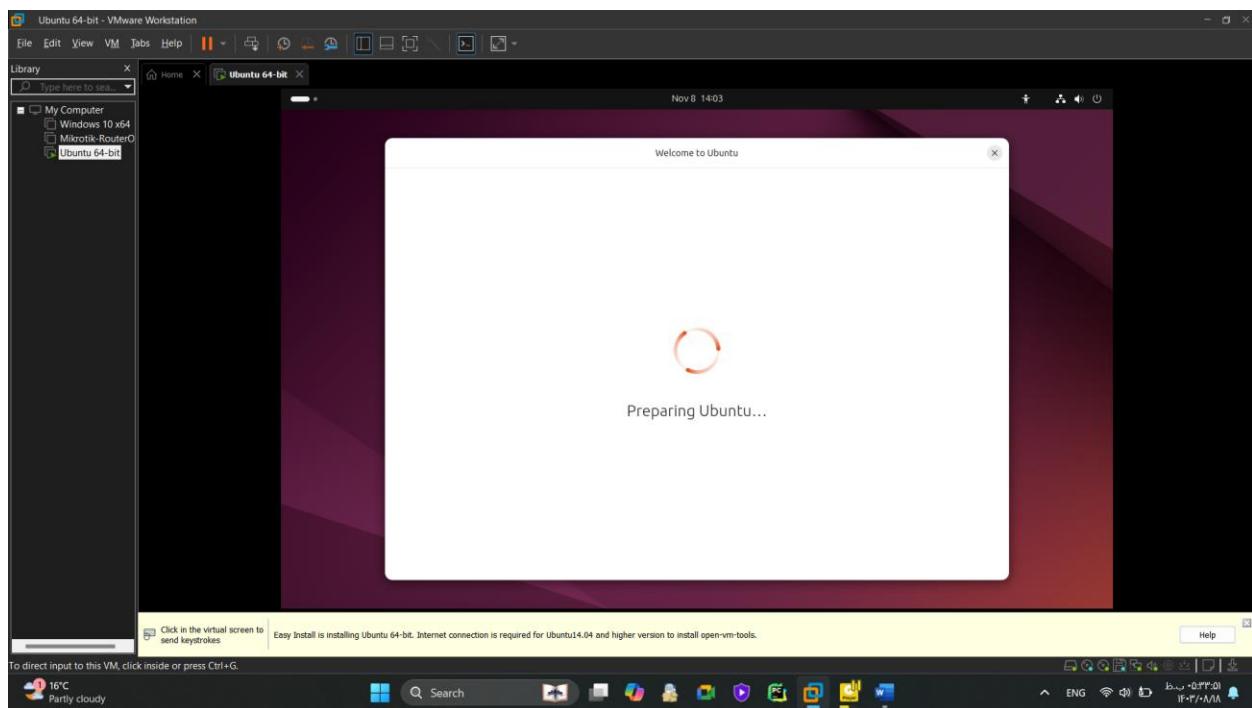
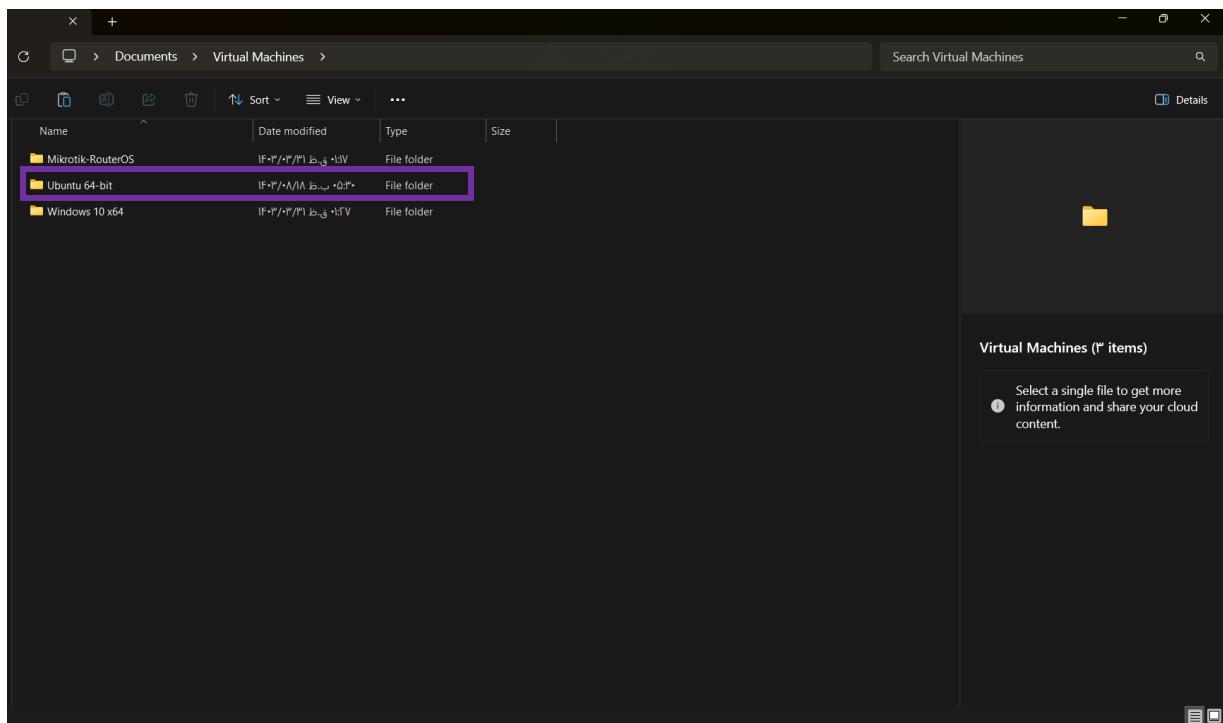
- Memory: 8 GB
- Processor: 2
- CD/DVD (SATA): Using file C:\Users\arefe\De...
- Network Adapter: NAT
- USB Controller: Present
- Display: Auto detect

The right window shows the following settings for the virtual machine:

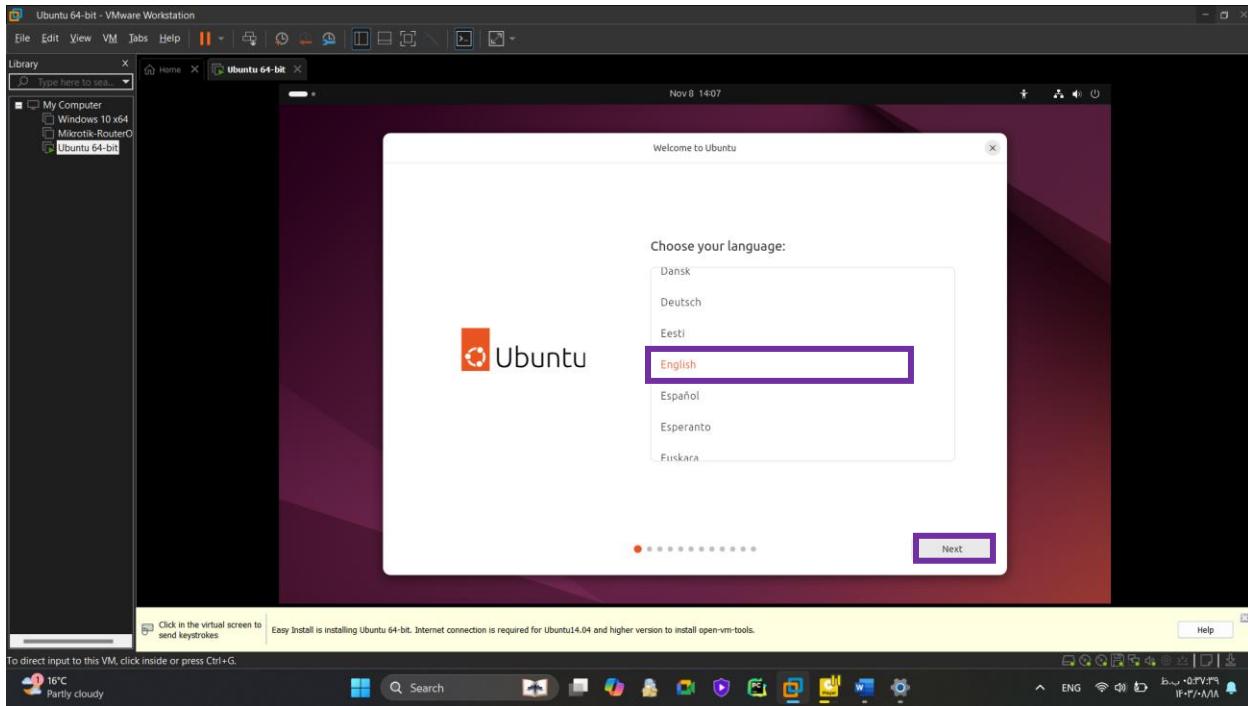
- Name: Ubuntu 64-bit
- Location: C:\Users\arefe\Documents\Virtual Machines\Ubuntu...
- Version: Workstation 17.5.x
- Operating System: Ubuntu 64-bit
- Hard Disk: 100 GB
- Memory: 8192 MB
- Network Adapter: NAT
- Other Devices: 2 CPU cores, CD/DVD, USB Controller, Sound Card

A purple box highlights the 'Power on this virtual machine after creation' checkbox.

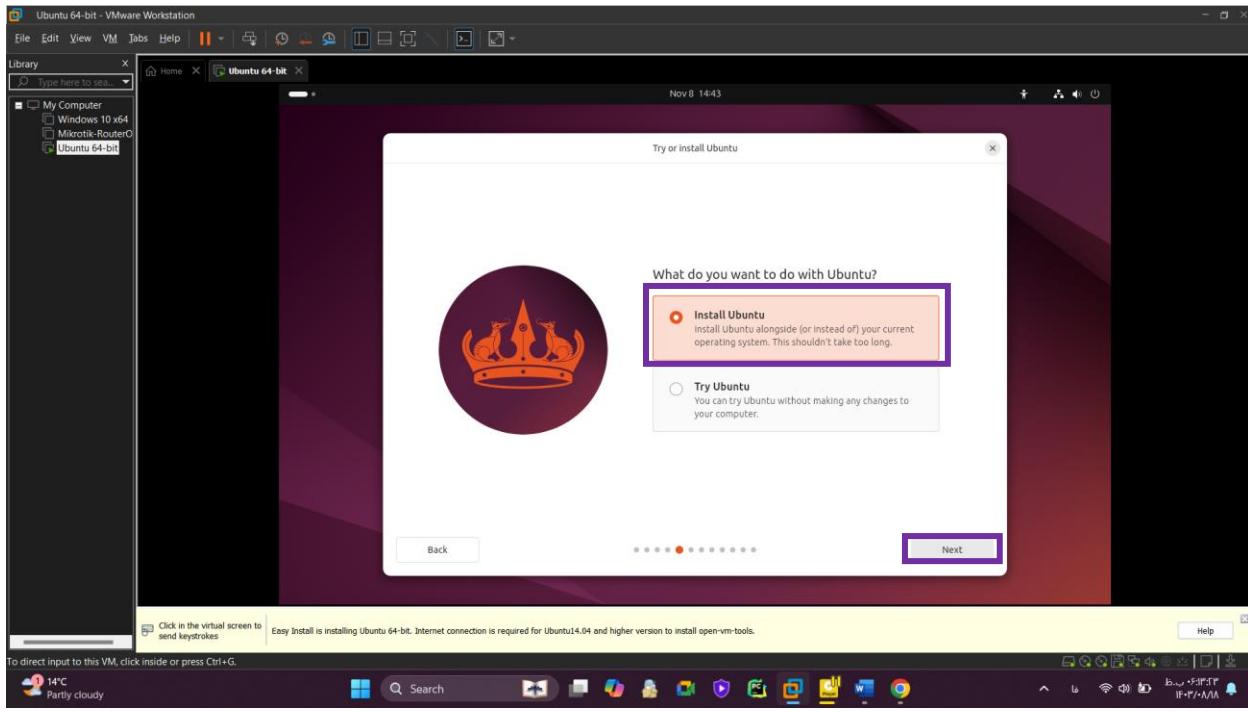
از اینجا هم میشه چک کرد که به درستی نصب شده است یا خیر

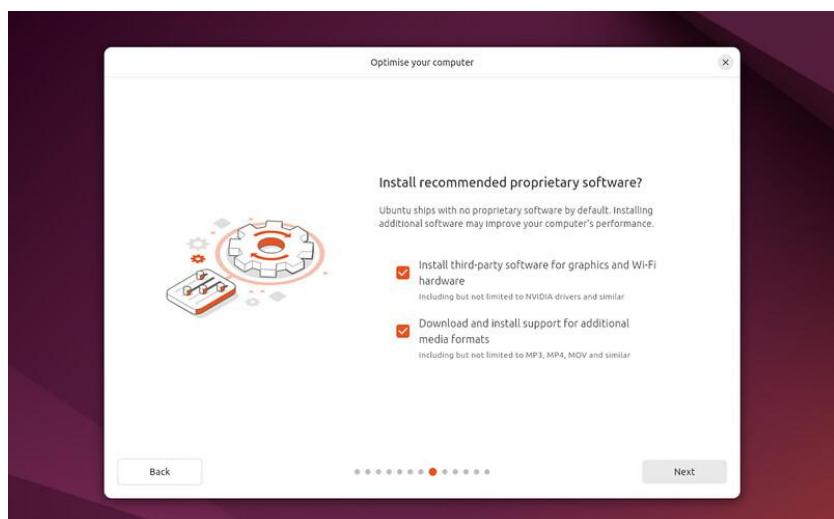
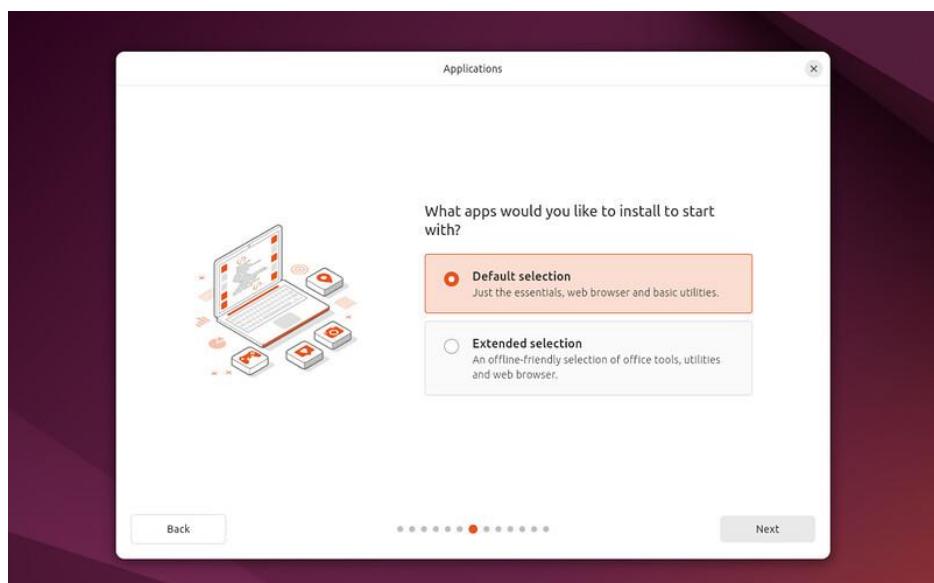
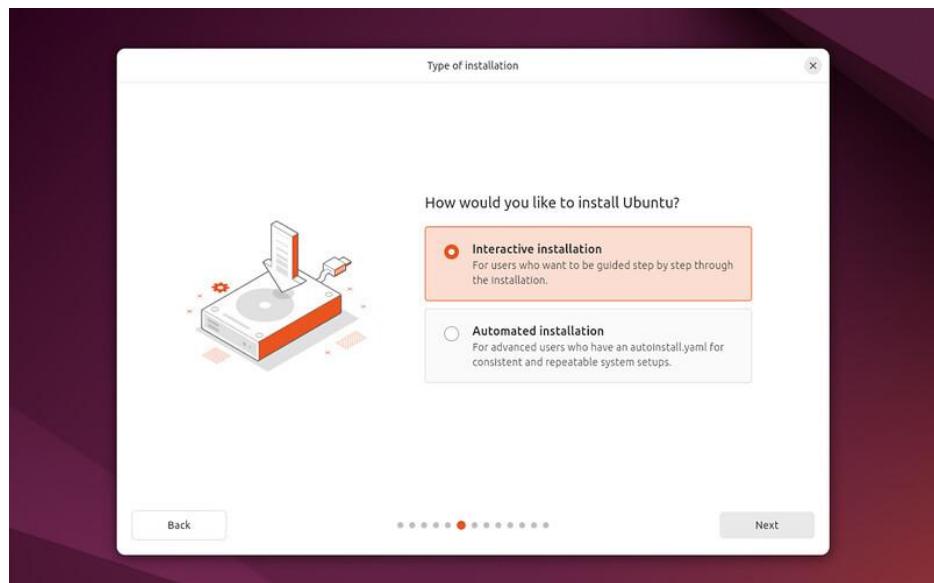


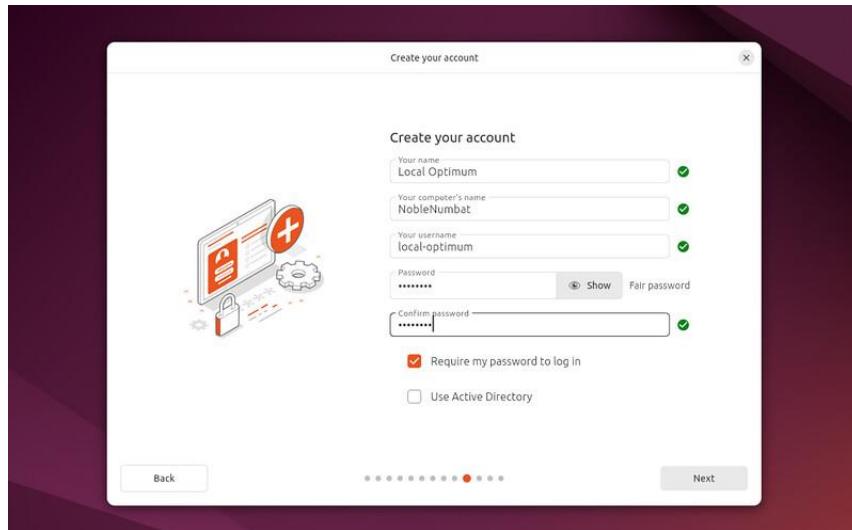
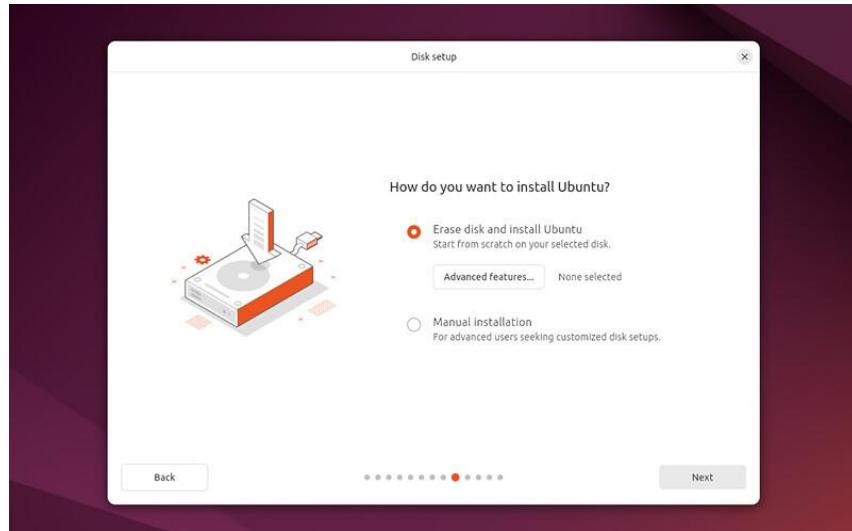
Open-vm-tools چیزیه که میشه یک فایل رو از ویندوز کپی کرد داخل لینوکس یا بر عکس. سیستم میشه full screen بشه. ۱. اینترنت حتما باید باشه ۲. ارور داد ۴۰۴ یا not found گذر از تحریم فعل شود بعد دوباره امتحان گردد.

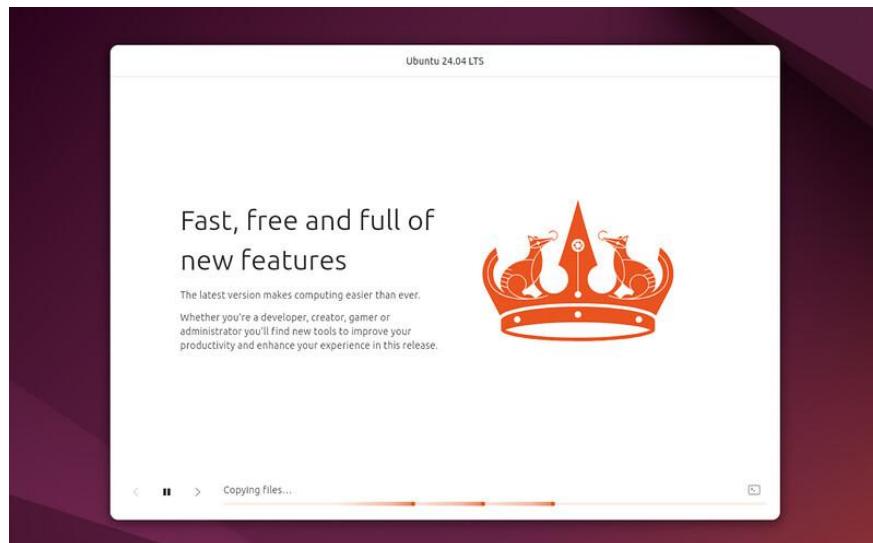
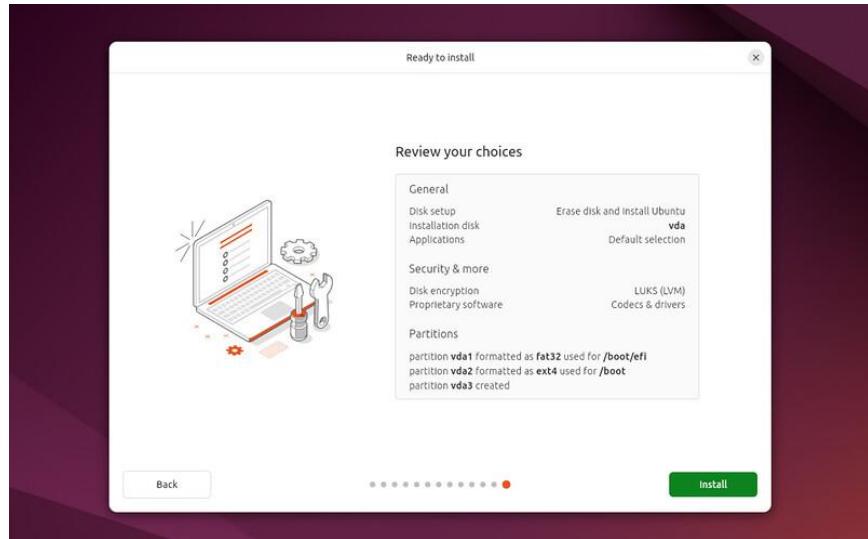


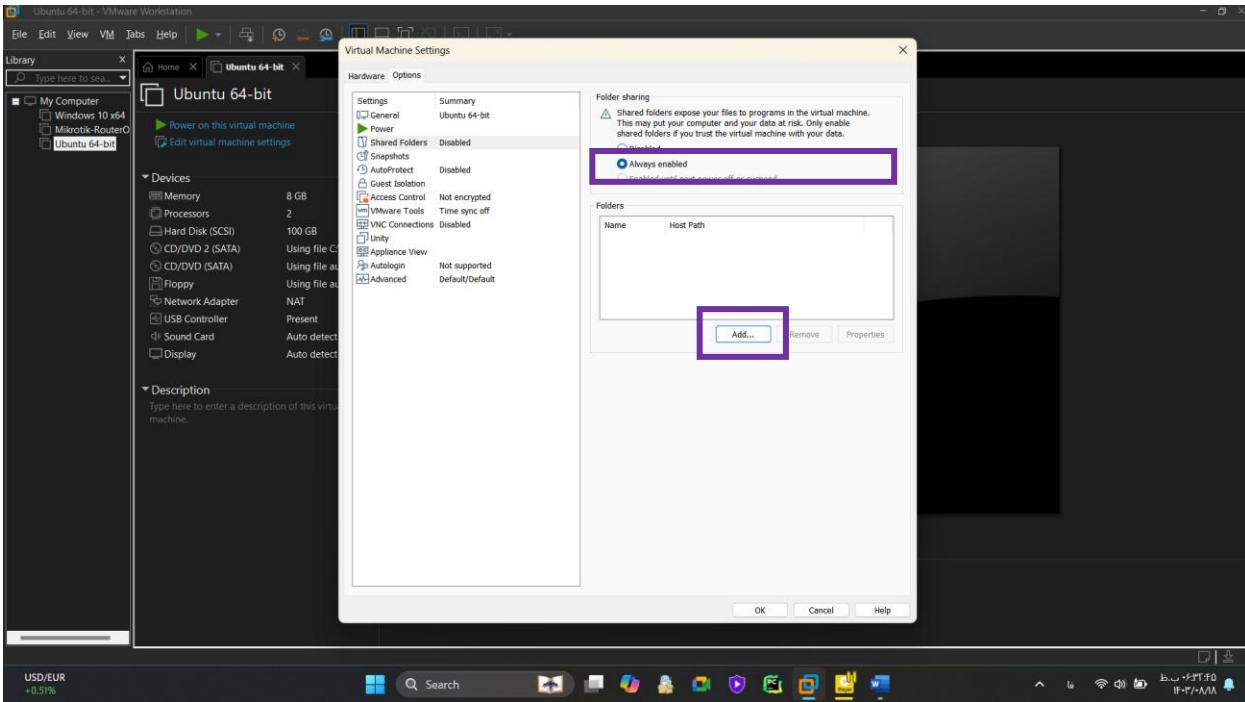
استفاده میکنی از genome





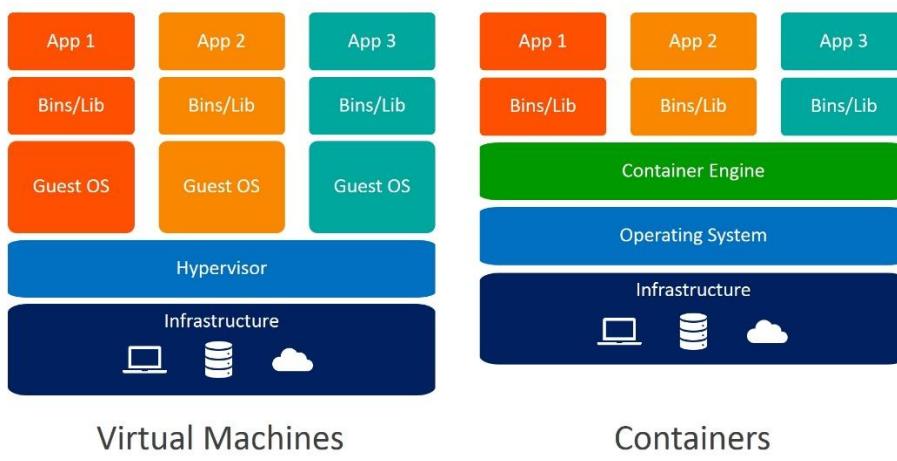






Docker

چیزی که عینا در product میبینیم در development هم ببینیم و به conflict نخوریم
تجربه کاری کانفیگ سرور یا لینوکس، از داکر روی محیط production میخوایم استفاده کنیم
کانفیگ بشه و به صورت داکر استفاده کنیم Lamp
پروژه جنگو با react رو پابلیش کنیم/ پروژه nodeJS پابلیش بشه



چه مراحتی در لینوکس داره
و در داکر اجرا بشه

داکر از لینوکس استفاده میکنه (داکر روی OS پیاده سازی میشه و هرگز از اون استفاده میکن)

یک محیط ایزوله برای OS ایجاد میکنه

و بخش های مختلف رو ران میکنه

کانتینر روی همون سرور

داکر نصب میشد و بخش های مختلف رو میسازه

در کانتینر OS مخصوص رو نداریم

داکر از کرنل استفاده میکنه

کرنل ارتباط سخت افزار با نرم افزار رو برقرار میکنه

لینوکس یک کرنل که روی سخت افزار سوار میشه و ابزارهای نرم افزار سوار میشه روی لینوکس(ارتباط بین سخت افزار با نرم افزار رو برقرار میکنه)

به جای اینکه برای هر محیط ایزوله یک کرنل نصب کنه از کرنل OS اصلی استفاده میکنه

اپلیکیشن ها روی داکر پیاده میشنه

VM ها حجم زیادی دارن (هرکدام OS خودش رو نصب میکنه و ازش استفاده میکنه)

Container docker خیلی سیک هستن

Kernel با ابزارهای مورد نیاز برای بالا امدن OS رو ندارن فقط اپ ها میان بالا و ران میشن

صرف منابع کمeh چون کرنل تکرار نمیشه

و سرعت کانتینر های داکر بالاتر

لوگوی داکر:

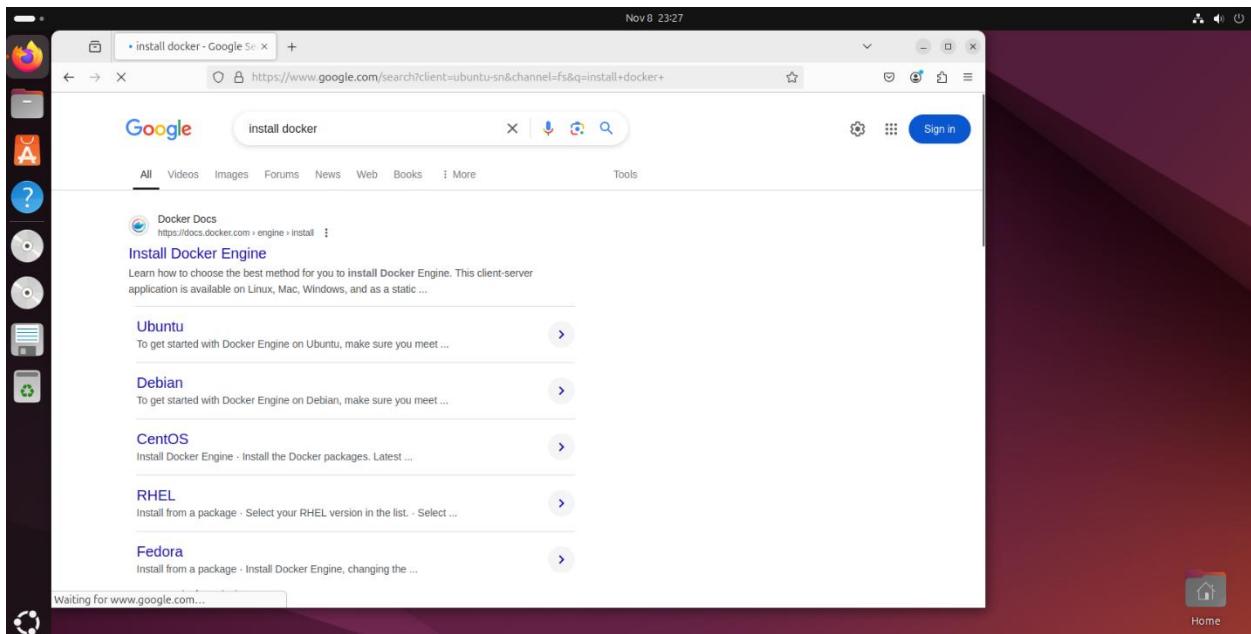
یک نهنگ که زیر آبه و بخش های زیرینش دیده نمیشه

همون کرنل

و اون مریع ها اپ های ما هستن که ما استفاده میکنیم برای ساختن محیط ایزوله خودمون



نصب داکر روی لینوکس:



A screenshot of a web browser displaying the Docker documentation for Ubuntu. The URL is https://docs.docker.com/engine/install/ubuntu/. The page is titled "Ubuntu | Docker Docs". It contains sections on "Uninstall old versions", "Prerequisites", "Firewall limitations", "OS requirements", and "Installation methods". A sidebar on the left shows navigation links for Docker Engine, Install, and other platforms like Docker Build, Docker Compose, and Docker Desktop. A code snippet at the bottom shows the command to uninstall conflicting packages: \$ for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do apt-get remove --purge \$pkg; done

Installation methods

You can install Docker Engine in different ways, depending on your needs:

- Docker Engine comes bundled with [Docker Desktop for Linux](#). This is the easiest and quickest way to get started.
- Set up and Install Docker Engine from [Docker's apt repository](#).
- [Install it manually](#) and manage upgrades manually.
- Use a [convenience script](#). Only recommended for testing and development environments.

Install using the apt repository

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker apt repository. Afterward, you can install and update Docker from the repository.

1. Set up Docker's apt repository.

```
# Add Docker's official GPG key:  
sudo apt-get update  
sudo apt-get install ca-certificates curl  
sudo install -m 0755 -d /etc/apt/keyrings  
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc  
sudo chmod a+r /etc/apt/keyrings/docker.asc  
  
# Add the repository to Apt sources:  
echo 'deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo "$VERSION_CODENAME") stable' | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

Installation methods

Docker Engine comes bundled with [Docker Desktop for Linux](#). This is the easiest and quickest way to get started.

- Set up and Install Docker Engine from [Docker's apt repository](#).
- [Install it manually](#) and manage upgrades manually.
- Use a [convenience script](#). Only recommended for testing and development environments.

Install using the apt repository

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker apt repository. Afterward, you can install and update Docker from the repository.

1. Set up Docker's apt repository.

```
# Add Docker's official GPG key:  
sudo apt-get update  
sudo apt-get install ca-certificates curl  
sudo install -m 0755 -d /etc/apt/keyrings  
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc  
sudo chmod a+r /etc/apt/keyrings/docker.asc  
  
# Add the repository to Apt sources:  
echo 'deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo "$VERSION_CODENAME") stable' | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "bt@bt-VMware-Virtual-Platform:~". The terminal output shows the following command sequence:

```
bt@bt-VMware-Virtual-Platform:~$ sudo apt-get remove docker docker-engine docker.io containerd runc
[sudo] password for bt:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package 'docker' is not installed, so not removed
E: Unable to locate package docker-engine
bt@bt-VMware-Virtual-Platform:~$ [[200~# Add Docker's official GPG key:
> sudo apt-get update
> sudo apt-get install ca-certificates curl
> sudo install -m 0755 -d /etc/apt/keyrings
> sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
> sudo chmod a+r /etc/apt/keyrings/docker.asc
>
> # Add the repository to Apt sources:
echo |
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu |
> $(. /etc/os-release && echo "SVERSION_CODENAME") stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
> ^C
bt@bt-VMware-Virtual-Platform:~$ sudo apt-get update
Warning: The unit file, source configuration file or drop-ins of apt-news.service changed on disk.
Run 'systemctl daemon-reload' to reload units.
Warning: The unit file, source configuration file or drop-ins of esm-cache.service changed on disk.
Run 'systemctl daemon-reload' to reload units.
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://ir.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://ir.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://ir.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://ir.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [624 kB]
```

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "bt@bt-VMware-Virtual-Platform:~". The terminal output shows the following command sequence:

```
Get:18 http://ir.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 2,224 kB in 8s (269 kB/s)
Reading package lists... Done
bt@bt-VMware-Virtual-Platform:~$ sudo apt-get install ca-certificates curl
E: Could not get lock /var/lib/dpkg/lock-frontend. It is held by process 6071 (aptd)
N: Be aware that removing the lock file is not a solution and may break your system.
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), is another process using it?
bt@bt-VMware-Virtual-Platform:~$ ^C
bt@bt-VMware-Virtual-Platform:~$ ps aux | grep apt
root      6071  1.6  1.3 211648 107976 ?        SNl   23:42  0:04 /usr/bin/python3 /usr/sbin/aptd
bt      14722  0.0  0.0 17812 2304 pts/0    S+   23:47  0:00 grep --color=auto apt
bt@bt-VMware-Virtual-Platform:~$ sudo apt-get install ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following NEW packages will be installed:
curl
0 upgraded, 1 newly installed, 0 to remove and 14 not upgraded.
Need to get 227 kB of archives.
After this operation, 534 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ir.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.4 [22
7 kB]
Fetched 227 kB in 1s (238 kB/s)
Selecting previously unselected package curl.
Reading database... 148974 files and directories currently installed.
Preparing to unpack .../curl_8.5.0-2ubuntu10.4_amd64.deb ...
Unpacking curl (8.5.0-2ubuntu10.4) ...
Setting up curl (8.5.0-2ubuntu10.4) ...
```

The screenshot shows a Linux desktop environment with a terminal window open in the background. The terminal window displays a command-line session for installing Docker on Ubuntu. The session includes commands like `sudo apt-get install docker` and `sudo docker run hello-world`. The desktop environment includes a dock with icons for various applications like a file manager, terminal, and browser.

```
Nov 8 23:57
bt@bt-VMware-Virtual-Platform:~ Processing triggers for man-db (2.12.0-4build2) ...
bt@bt-VMware-Virtual-Platform:~ sudo install -m 0755 -d /etc/apt/keyrings
bt@bt-VMware-Virtual-Platform:~ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | /etc/apt/keyrings/docker.asc
bt@bt-VMware-Virtual-Platform:~ sudo chmod a+r /etc/apt/keyrings/docker.asc
bt@bt-VMware-Virtual-Platform:~ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$ /etc/os-release && echo \"$VERSION_CODENAME\" stable" | \ 
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
bt@bt-VMware-Virtual-Platform:~ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:2 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:3 https://download.docker.com/linux/ubuntu/stable amd64 Packages [15.3 kB]
Hit:4 http://ir.archive.ubuntu.com/ubuntu noble InRelease
Hit:5 http://ir.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:6 http://ir.archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 64.2 kB in 5s (14.1 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email
  git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras
  docker-compose-plugin git liberror-perl libslirp0 pigz slirp4netns
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email
  git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras
  docker-compose-plugin git liberror-perl libslirp0 pigz slirp4netns
```

This screenshot is nearly identical to the one above, showing the same terminal session and desktop environment. The terminal window continues the Docker installation process, showing the download of the 'hello-world' image and its execution.

```
Nov 8 23:57
bt@bt-VMware-Virtual-Platform:~ Setting up git (1:2.43.0-1ubuntu7.1) ...
bt@bt-VMware-Virtual-Platform:~ Processing triggers for man-db (2.12.0-4build2) ...
bt@bt-VMware-Virtual-Platform:~ Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
bt@bt-VMware-Virtual-Platform:~ sudo docker run hello-world
bt@bt-VMware-Virtual-Platform:~ Unable to find image 'hello-world:latest' locally
bt@bt-VMware-Virtual-Platform:~ latest: Pulling from library/hello-world
bt@bt-VMware-Virtual-Platform:~ clec1eb5944: Pull complete
bt@bt-VMware-Virtual-Platform:~ Digest: sha256:d211f485f2dd1dee407a080973c8f129f00d54604d2c90732e8e320e5038a0348
bt@bt-VMware-Virtual-Platform:~ Status: Downloaded newer image for hello-world:latest
bt@bt-VMware-Virtual-Platform:~ Hello from Docker!
bt@bt-VMware-Virtual-Platform:~ This message shows that your installation appears to be working correctly.
bt@bt-VMware-Virtual-Platform:~ To generate this message, Docker took the following steps:
bt@bt-VMware-Virtual-Platform:~ 1. The Docker client contacted the Docker daemon.
bt@bt-VMware-Virtual-Platform:~ 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
bt@bt-VMware-Virtual-Platform:~ (and64)
bt@bt-VMware-Virtual-Platform:~ 3. The Docker daemon created a new container from that image which runs the
bt@bt-VMware-Virtual-Platform:~ executable that produces the output you are currently reading.
bt@bt-VMware-Virtual-Platform:~ 4. The Docker daemon streamed that output to the Docker client, which sent it
bt@bt-VMware-Virtual-Platform:~ to your terminal.
bt@bt-VMware-Virtual-Platform:~ To try something more ambitious, you can run an Ubuntu container with:
bt@bt-VMware-Virtual-Platform:~ $ docker run -it ubuntu bash
bt@bt-VMware-Virtual-Platform:~ Share images, automate workflows, and more with a free Docker ID:
bt@bt-VMware-Virtual-Platform:~ https://hub.docker.com/
bt@bt-VMware-Virtual-Platform:~ For more examples and ideas, visit:
bt@bt-VMware-Virtual-Platform:~ https://docs.docker.com/get-started/
```

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "bt@bt-VMware-Virtual-Platform:~". It displays the following text:

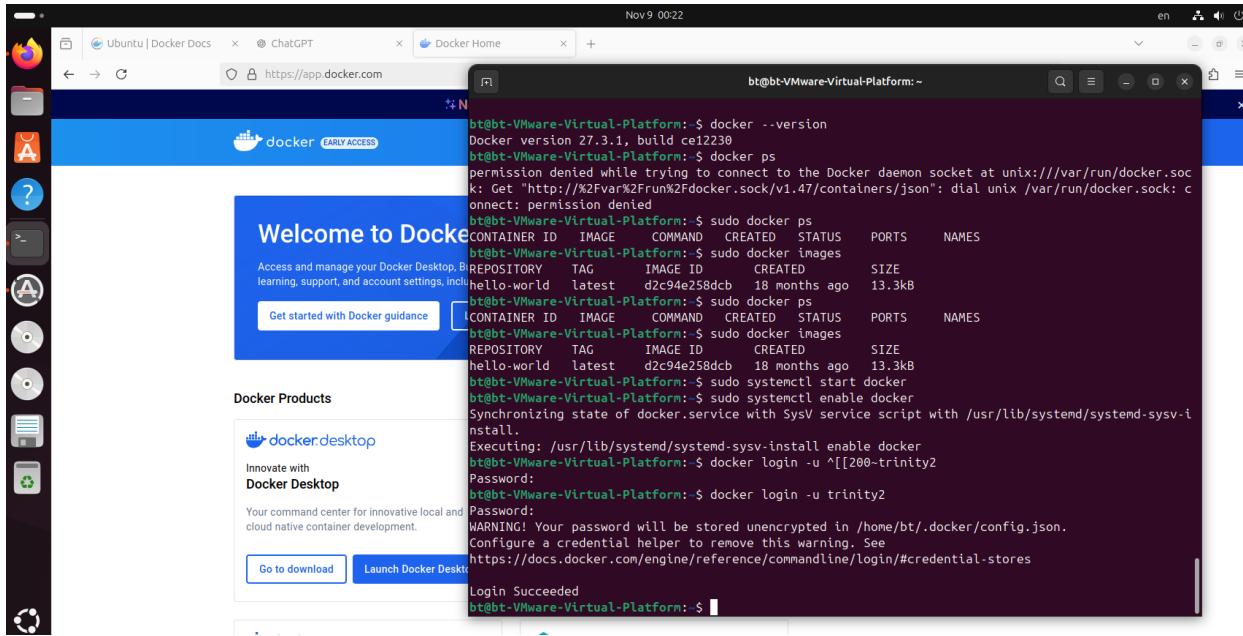
```
$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
3. Verify that the installation is successful
$ sudo docker run hello-world
Hello from Docker!
This command downloads a Docker image and runs the executable that produces the output you are currently reading.
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (andreas)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.
You have now successfully installed Docker on this machine.
Tip
Receiving errors when trying to run Docker commands? The docker user group exists. Continue to Linux optional configuration steps.
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
Install from a package
```

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "bt@bt-VMware-Virtual-Platform:~". It displays the following text:

```
(andreas)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
You have now successfully installed Docker on this machine.
Tip
Receiving errors when trying to run Docker commands? The docker user group exists. Continue to Linux optional configuration steps.
Upgrade Docker Engine
To upgrade Docker Engine, follow the instructions in the Docker Engine documentation.
Install from a package
```

At the bottom of the terminal window, there is a scrollable log output:

```
bt@bt-VMware-Virtual-Platform:~$ docker --version
Docker version 27.3.1, build ce12230
bt@bt-VMware-Virtual-Platform:~$ docker ps
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock
error: Get "http://127.0.0.1:2375/json": dial unix /var/run/docker.sock: connect: permission denied
bt@bt-VMware-Virtual-Platform:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest d2c94e258dcb 18 months ago 13.3kB
bt@bt-VMware-Virtual-Platform:~$ sudo docker images
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest d2c94e258dcb 18 months ago 13.3kB
bt@bt-VMware-Virtual-Platform:~$
```



از هر IP میشه 100 image رو دانلود کرد از سرور داکر

محدودیت دانلود اینمیج ها در هر ۶ ساعت افزایش داده شده احتمال به مشکل خوردن کمتر میشه. با vpn راحتتر میشه اینمیج ها دانلود میشه و از داکر استفاده میکنیم.

نصب وب سرور، PHP و پایگاه داده (LAMP Stack)

برای شبیه سازی عملکرد WAMP (Linux, Apache, MySQL, PHP) در لینوکس، می توانید از LAMP (Linux, Apache, MySQL, PHP) استفاده کنید.

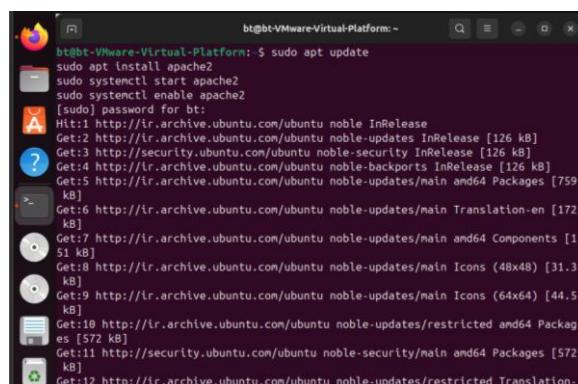
Apache: نصب

sudo apt update

sudo apt install apache2

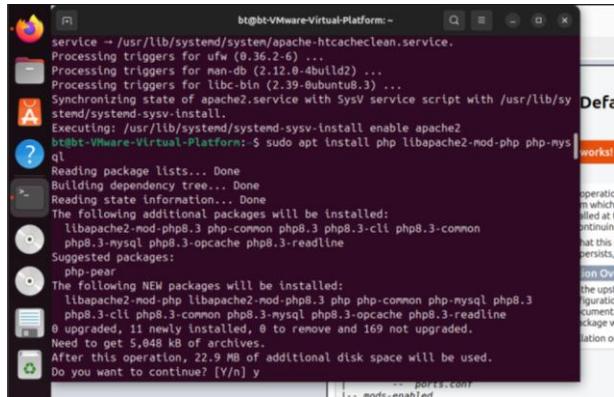
sudo systemctl start apache2

sudo systemctl enable apache2



نصب PHP:

```
sudo apt install php libapache2-mod-php php-mysql
```



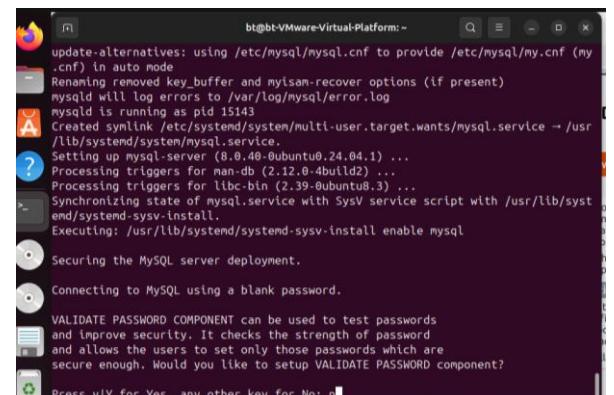
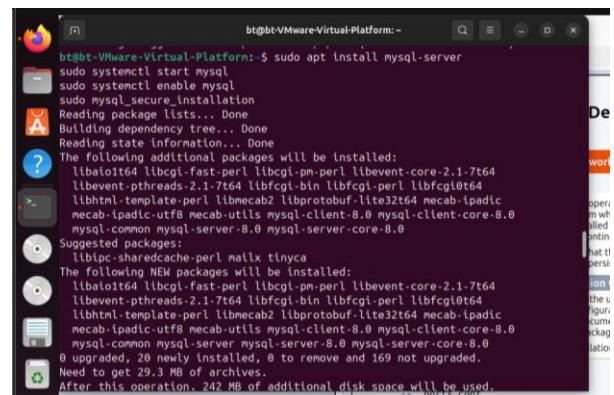
نصب MySQL

```
sudo apt install mysql-server
```

sudo systemctl start mysql

```
sudo systemctl enable mysql
```

sudo mysql_secure_installation



```

bt@bt-VMware-Virtual-Platform:~ 'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : n

A ... skipping.
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No)
: n

... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : n

... skipping.
All done!
ht@bt-VMware-Virtual-Platform:~ $ 

```

تست PHP:

یک فایل تست ایجاد کنید:

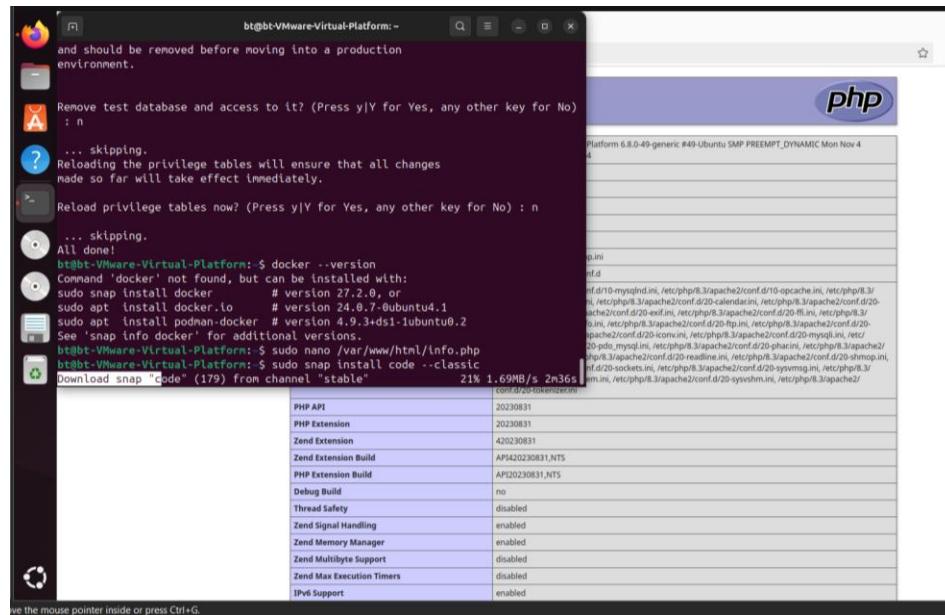
```
sudo nano /var/www/html/info.php
```

در فایل، این خط را وارد کنید:

```
<?php phpinfo(); ?>
```

مرورگر را باز کرده و آدرس زیر را وارد کنید:

<http://localhost/info.php>



The screenshot shows a terminal window titled "bt@bt-VMware-Virtual-Platform: ~". It contains two tabs: "GNU nano 7.2" and "/var/www/html/info.php". The "/var/www/html/info.php" tab displays the output of a PHP script, specifically the `<?php phpInfo(); ?>` command. The output includes various PHP configuration settings and extensions. The terminal has a dark theme with white text and a black background. A vertical toolbar on the left contains icons for file operations like Open, Save, and Cut.

نصب اپزارهای ویرایش کد

گزینه‌های پیشنهادی:

Visual Studio Code: •

۱. دانلود و نصب:

sudo snap install code --classic

افزونه‌های لازم:

PHP Intelephense •

(در صورت نیاز) Docker •

MySQL •

guidelines. We may choose to rename function arguments when necessary in order to improve the Laravel codebase. Therefore, using named arguments when calling Laravel methods should be done cautiously and with the understanding that the parameter names may change in the future.

Support Policy

For all Laravel releases, bug fixes are provided for 18 months and security fixes are provided for 2 years. For all additional libraries, including Lumen, only the latest major release receives bug fixes. In addition, please review the database versions [supported by Laravel](#).

Version	PHP (*)	Release	Bug Fixes Until	Security Fixes Until
9	8.0 - 8.2	February 8th, 2022	August 8th, 2023	February 6th, 2024
10	8.1 - 8.3	February 14th, 2023	August 6th, 2024	February 4th, 2025
11	8.2 - 8.4	March 12th, 2024	September 3rd, 2025	March 12th, 2026
12	8.2 - 8.4	Q1 2025	Q3 2026	Q1 2027

■ End of life ■ Security fixes only
(*) Supported PHP versions

Laravel 11

ایجاد پروژه کرد لیست برنامه روزانه

```

ERROR 1045 (28000): Access denied for user
'b'@'localhost' (using password: NO)
bt@bt-VMware-Virtual-Platform: ~ $ mysql --version
mysql Ver 8.0.48-0ubuntu0.24.04.1 for Lin
ux on x86_64 ((Ubuntu))
bt@bt-VMware-Virtual-Platform: ~ $ php -v
PHP 8.3.6 (cli) (built: Dec 2 2024 12:36:
18) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Tec
hnologies
with Zend OPcache v8.3.6, Copyright (c
), by Zend Technologies
bt@bt-VMware-Virtual-Platform: ~ $ composer
-v

```

PHP Version 8.3.6

System	Linux bt-VMware-Virtual-Platform 6.8.0-51-generic #52-Ubuntu SMP PREEMPT_DYNAMIC Thu Dec 5 13:09:44 UTC 2024 x86_64
Build Date	Dec 2 2024 12:36:18
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqlind.ini, /etc/php/8.3/apache2/conf.d/10-epochache.ini, /etc/php/8.3/ apache2/conf.d/10-gd.ini, /etc/php/8.3/apache2/conf.d/15-xml.ini, /etc/php/8.3/apache2/conf.d/20-bz2.ini, /etc/php/8.3/apache2/conf.d/20-calendariini, /etc/php/8.3/apache2/conf.d/20-type.ini, /etc/php/8.3/ apache2/conf.d/20-zip.ini, /etc/php/8.3/apache2/conf.d/20-zipfpm.ini, /etc/php/8.3/apache2/conf.d/20- zlib.ini, /etc/php/8.3/apache2/conf.d/20-zmfilter.ini, /etc/php/8.3/apache2/conf.d/20-zmfilterfpm.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gd-gd.ini, /etc/php/8.3/apache2/conf.d/20- gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-lm.ini, /etc/php/8.3/apache2/ conf.d/20-mbstring.ini, /etc/php/8.3/apache2/conf.d/20-mcrypt.ini, /etc/php/8.3/apache2/conf.d/20- myqlind.ini, /etc/php/8.3/apache2/conf.d/20-pdo.ini, /etc/php/8.3/apache2/conf.d/20-pdo_mysqli.ini, /etc/php/8.3/apache2/conf.d/20-pdo_sqlite.ini, /etc/php/8.3/apache2/conf.d/20-pdo_sqlite3.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20- simplexml.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.3/apache2/conf.d/20-sysvsem.ini, /etc/php/8.3/apache2/conf.d/20-tokenizerini, /etc/php/8.3/apache2/conf.d/20- xmlreaderini, /etc/php/8.3/apache2/conf.d/20-xmlwriterini, /etc/php/8.3/apache2/conf.d/20-xslini, /etc/ php/8.3/apache2/conf.d/20-zip.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API20230831_ANTS
PHP Extension Build	API20230831_ANTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled

Ubuntu 64-bit - VMware Workstation

```
bt@bt-VMware-Virtual-Platform: ~ composer -v
Composer version 2.7.1 2024-02-09 15:26:28
Usage:
  command [options] [arguments]
Options:
  -h, --help           Display help for the given command. When no command is given display help for the list command
  and
  -q, --quiet          Do not output any message
  -V, --version         Display this application version
  --ansi|--no-ansi     Force (or disable --no-ansi) ANSI output
  -n, --no-interaction  Do not ask any interactive question
  --profile            Display timing and memory usage information
  --no-plugins          Whether to disable plugins.
  --no-scripts          Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache            Prevent use of the cache
PHP API
20230831
PHP Extension
20230831
Zend Extension
420230831
Zend Extension Build
API420230831_ANTS
PHP Extension Build
API20230831_ANTS
Debug Build
no
Thread Safety
disabled
Zend Signal Handling
enabled
```

To direct input to this VM, click inside or press Ctrl+G.

Ubuntu 64-bit - VMware Workstation

```
bt@bt-VMware-Virtual-Platform: ~ composer creat-project laravel/laravel crud
Command "creat-project" is not defined.

Do you want to run "create-project" instead? (yes/no) [no]:
> no

bt@bt-VMware-Virtual-Platform: ~ composer create-project --prefer-dist laravel/laravel product-management^C
bt@bt-VMware-Virtual-Platform: ~ laravel new first-project
laravel: command not found
bt@bt-VMware-Virtual-Platform: ~ composer create-project --prefer-dist laravel/laravel first-project
Creating a 'laravel/laravel' project at './first-project'
Installing laravel/laravel (v11.5.0)
- Downloading laravel/laravel (v11.5.0)
- Installing laravel/laravel (v11.5.0): Extracting archive
```

storefront

test

test.txt

test.sh

Home

daily planner

To direct input to this VM, click inside or press Ctrl+G.

Ubuntu 64-bit - VMware Workstation

File Edit View VM Tabs Help | Jan 9 19:51

Home Ubuntu 64-bit

```
bt@bt-VMware-Virtual-Platform:~/first-project
> @php artisan vendor:publish --tag=laravel-assets --ansi --force
INFO No publishable resources for tag [laravel-assets].
No security vulnerability advisories found.
> @php artisan key:generate --ansi
INFO Application key set successfully.
> @php -r "file_exists('database/database.sqlite') || touch('database/database.sqlite');"
> @php artisan migrate --graceful --ansi
INFO Preparing database.
Creating migration table ..... 28.21ms DONE
INFO Running migrations.
0001_01_01_000000 create_users_table ..... 39.64ms DONE
0001_01_01_000001 create_cache_table ..... 14.24ms DONE
0001_01_01_000002 create_jobs_table ..... 38.52ms DONE
bt@bt-VMware-Virtual-Platform: $ cd first-project
bt@bt-VMware-Virtual-Platform:~/first-project$ code .
```

To direct input to this VM, click inside or press Ctrl+G.

Ubuntu 64-bit - VMware Workstation

File Edit View VM Tabs Help | Jan 9 2027

Home Ubuntu 64-bit

edit.blade.php - Cloud_Computing_Project - Visual Studio Code

EXPLORER

- GROUP 1
 - create.blade.php
 - index.blade.php
 - edit.blade.php
- GROUP 2
 - 2025_01_08_131737_create_tasks_table.php

resources > views > tasks > edit.blade.php > Form

```
<div class="alert alert-danger">
  <ul>
    @foreach ($errors->all() as $error)
      <li>{{ $error }}</li>
    @endforeach
  </ul>
</div>
@endif
```

A space-separated list of the classes of the element. Classes allows CSS and JavaScript to select and access specific elements via the class selectors or functions like the method

```
<form act @csrf @meth Document.getElementsByClassName() .
```

MDN Reference

```
<div class="row">
  <div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
      <strong>Name:</strong>
      <input type="text" name="name" />
    </div>
  </div>
</div>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

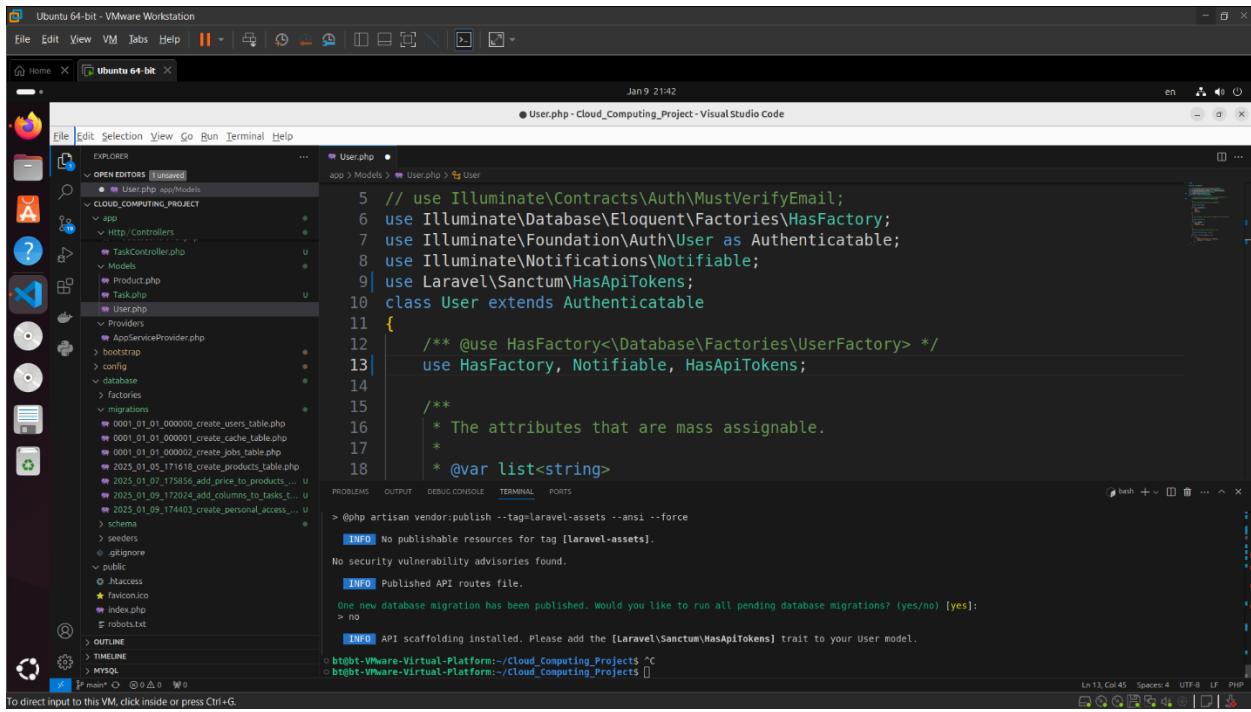
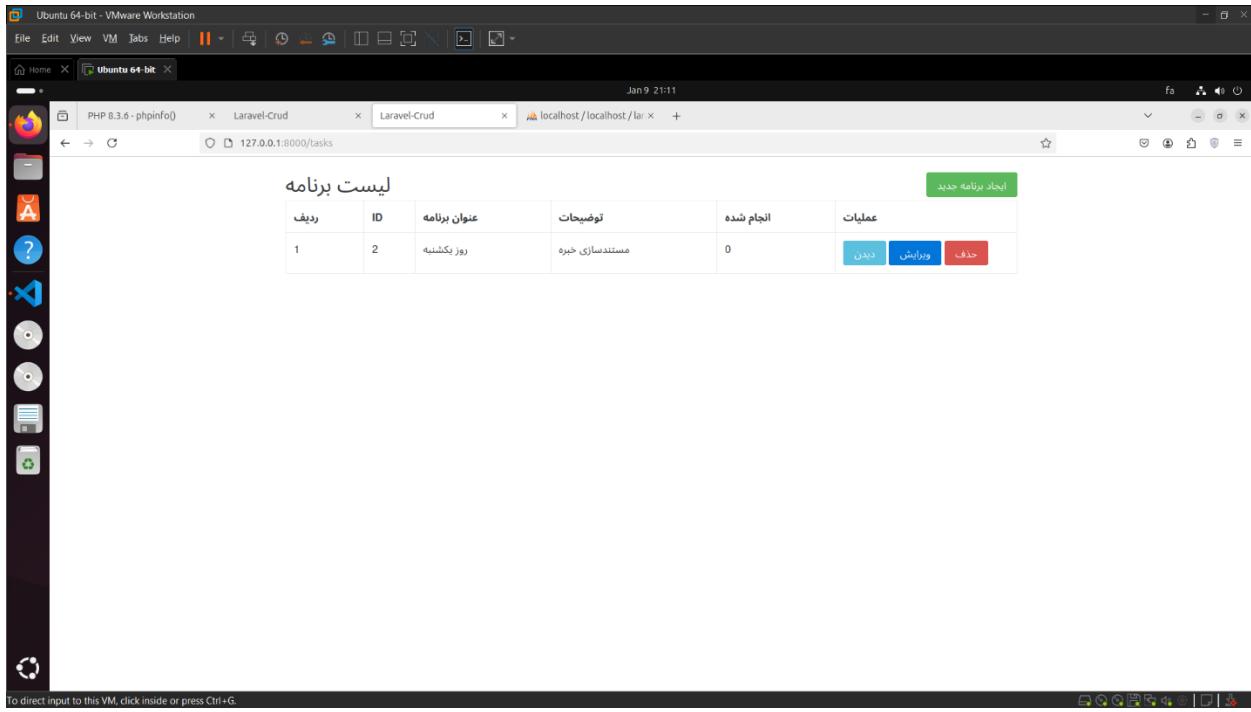
2025-01-09 20:08:46 /products

2025_01_08_131737_create_tasks_table.php > class > up > c

```
7 return new class extends Migration
{
  /**
   * Run the migrations.
   */
  public function up(): void
  {
    Schema::create('tasks',
      $table->id();
      $table->string('title');
      $table->text('description');
      $table->boolean('completed');
      $table->timestamps();
    );
  }

  /**
   * Reverse the migrations.
   */
  public function down(): void
  {
```

Ln 28, Col 30 Spaces: 4 UTF-8 LF PHP



The screenshot shows a Visual Studio Code interface running on an Ubuntu 64-bit - VMware Workstation. The workspace is titled "api.php - Cloud_Computing_Project - Visual Studio Code".

File Explorer: Shows the project structure under "CLOUD_COMPUTING_PROJECT".

Editor: Displays the file "api.php" with the following code:

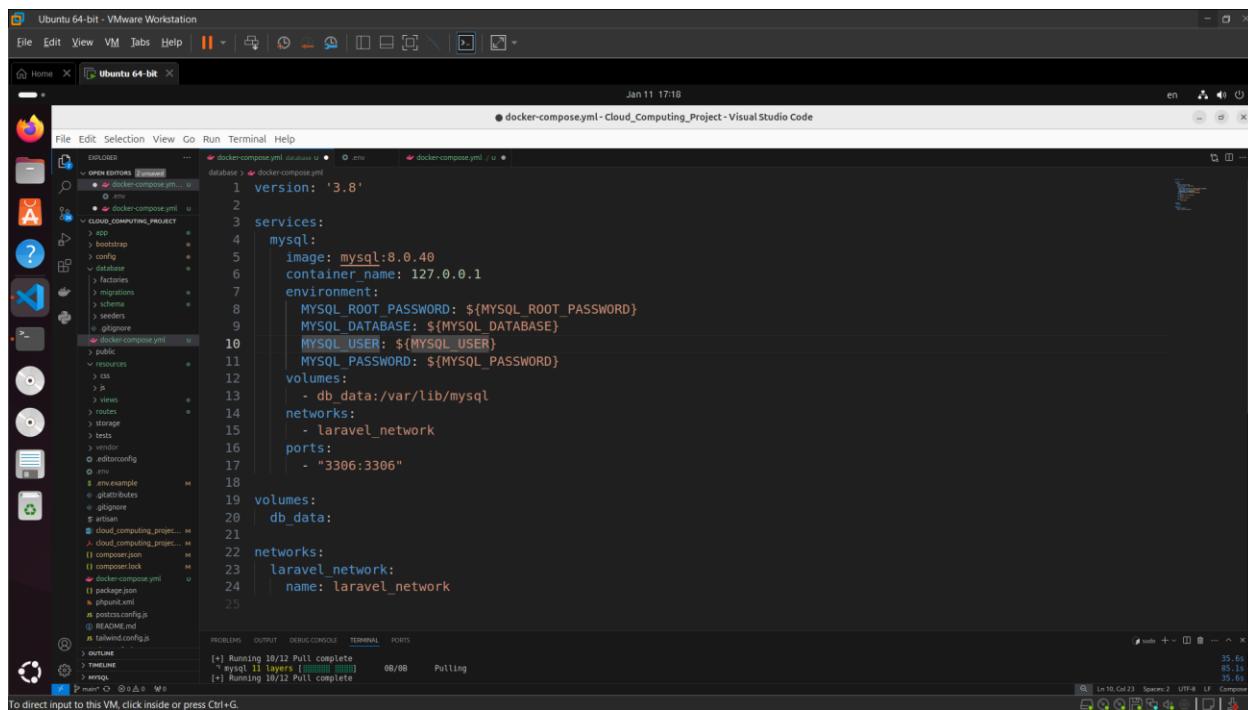
```
routes > api.php > ...
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 Route::get('/user', function (Request $request) {
7     return $request->user();
8 })->middleware('auth:sanctum');
```

Terminal:

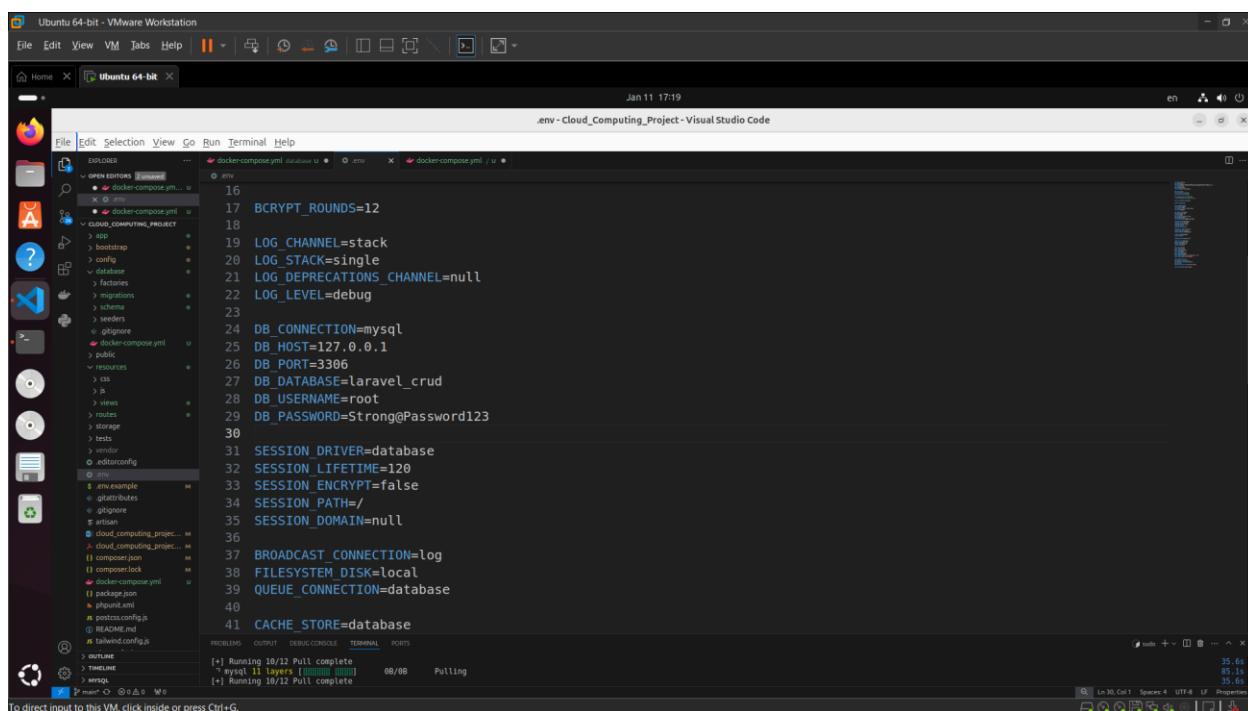
```
INFO | Published API routes file.
One new database migration has been published. Would you like to run all pending database migrations? (yes/no) [yes]:> no
INFO | API scaffolding installed. Please add the [Laravel\Sanctum\HasApiTokens] trait to your User model.
o btbt@VMware-Virtual-Platform:~/Cloud_Computing_Projects^C
o btbt@VMware-Virtual-Platform:~/Cloud_Computing_Projects php artisan migrate
INFO | Running migrations.
2025_01_09_174403 create personal access tokens table ..... 113.04ms DONE
o btbt@VMware-Virtual-Platform:~/Cloud_Computing_Projects ^C
```

Status Bar: Shows "Ln 9 Col 1 Spaces: 4 UTF-8 LF".

دامریز کردن پروژه و دیتابیس پروژه با شبکه خارجی کانتینرها



```
version: '3.8'
services:
  mysql:
    image: mysql:8.0.40
    container_name: 127.0.0.1
    environment:
      MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
      MYSQL_DATABASE: ${MYSQL_DATABASE}
      MYSQL_USER: ${MYSQL_USER}
      MYSQL_PASSWORD: ${MYSQL_PASSWORD}
    volumes:
      - db_data:/var/lib/mysql
    networks:
      - laravel_network
    ports:
      - "3306:3306"
volumes:
  db_data:
networks:
  laravel_network:
    name: laravel_network
```



```
BCRYPT_ROUNDS=12
LOG_CHANNEL=stack
LOG_STACK=single
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel_crud
DB_USERNAME=root
DB_PASSWORD=Strong@Password123

SESSION_DRIVER=database
SESSION_LIFETIME=120
SESSION_ENCRYPT=false
SESSION_PATH/
SESSION_DOMAIN=null

BROADCAST_CONNECTION=log
FILESYSTEM_DISK=local
QUEUE_CONNECTION=database

CACHE_STORE=database
```

The screenshot shows a VMware Workstation interface with an Ubuntu 64-bit VM running. Inside the VM, Visual Studio Code is open, displaying a Docker Compose file named `docker-compose.yml`. The file defines a service named `app` which builds from the current directory, uses a Dockerfile named `Dockerfile`, and runs under the container name `laravel_app`. It maps port `8000:80` and connects to a network named `laravel_network`. The environment variables are set to connect to a database host at `127.0.0.1` on port `3306`, with the database name, user, and password all set to `laravel`. The `networks:` section defines the `laravel_network` as external. Below the code editor, the terminal tab shows the output of a `docker pull` command for MySQL and PostgreSQL, indicating they have been successfully pulled.

```
version: '3.8'
services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
      container_name: laravel_app
    volumes:
      - ./var/www/html
    ports:
      - "8000:80"
    networks:
      - laravel_network
  environment:
    DB_HOST: 127.0.0.1
    DB_PORT: 3306
    DB_DATABASE: ${DB_DATABASE}
    DB_USERNAME: ${DB_USERNAME}
    DB_PASSWORD: ${DB_PASSWORD}
networks:
  laravel_network:
    external: true
```

To direct input to this VM, click inside or press Ctrl+G.

The screenshot shows a VMware Workstation interface with an Ubuntu 64-bit VM running. Inside the VM, Visual Studio Code is open, displaying a Docker Compose file. The file defines four MySQL services (mysql1 through mysql4) with various configurations like BCRYPT_ROUNDS and LOG_CHANNEL. The terminal window shows the execution of docker-compose up -d, which fails due to permission denied errors when trying to bind to the docker.sock socket. The logs also show MySQL instances starting up on ports 33060, 33061, 33062, and 33063. A status bar at the bottom right provides system information.

The screenshot shows a VMware Workstation window with an Ubuntu 64-bit VM running. The terminal window in VS Code displays the output of a Docker build command, showing multiple errors related to failed file copy operations due to network issues. The errors mention failed reads from https://production.cloudflare.docke... and TLS handshake timeouts.

```
Jan 11 19:45
dockerfile - Cloud_Computing_Project - Visual Studio Code

File Edit Selection View Go Run Terminal Help
File Edit Selection View Go Run Terminal Help
Dockerfile docker-compose.yml docker-compose.yml docker-compose.yml
14
15 # Copy the application code
16 COPY .
17
18 # Install Composer
19 COPY --from=composer:latest /usr/bin/composer /usr/bin/composer
20

> [app internal] load metadata for docker.io/library/php:8.2-apache:
> failed to solve: solved: failed to solve metadata for docker.io/library/php:8.2-apache: failed to copy: httpReaderFailed: failed open: failed to do request: Get "https://production.cloudflare.docke...
r://com/registry/v2/docker/registry/v2/blobs/sha256/4c46926ccf88ac0f0818a15e80d36a8cc818e0dcbebe58b6ccfd8c8130a/data?verity=173614559-IXAPIC2vC1a0Bnnz2BPMq1qta8J30": net/http: TLS handshake timeout
btsp@VMware-Virtual-Machine:/Cloud_Computing_Project$ sudo docker-compose up -d
19
19.0.2-2025-01-11-19-45-17 (773)
[1]: 19.0.2-2025-01-11-19-45-17 (773)
[1]: docker-compose
[1]: => snu256:1cf9f800d8a24931f7dd9ec730891ee49e112a179bbdbd216989ae1830f775 3.39B
[1]: => sha256:1f1e346996e296de4fa584de05676e3748b7315e06d1476c24403d592134f9
[1]: => snu256:1cf9f800d8a24931f7dd9ec730891ee49e112a179bbdbd216989ae1830f775 3.39B
[1]: => extracting sha256:1f1e346996e296de4fa584de05676e3748b7315e06d1476c24403d592134f9
[1]: => 0.4s
[1]: => snu256:1cf9f800d8a24931f7dd9ec730891ee49e112a179bbdbd216989ae1830f775 941B
[1]: => extracting sha256:1f1e346996e296de4fa584de05676e3748b7315e06d1476c24403d592134f9
[1]: => 0.2s
[1]: => sha256:1cf9f800d8a24931f7dd9ec730891ee49e112a179bbdbd216989ae1830f775 217B
[1]: => 200.7s
[1]: => extracting sha256:7d0bf1f53646b046d93c5ff1118c3b7f5193861954d1e7e7526308755e08 941B
[1]: => sha256:1cf9f800d8a24931f7dd9ec730891ee49e112a179bbdbd216989ae1830f775 217B
[1]: => 217B
[1]: => extracting sha256:7d0bf1f53646b046d93c5ff1118c3b7f5193861954d1e7e7526308755e08 941B
[1]: => 217B
[1]: => sha256:10778025f313aa04316e050801a397160fb718633d5498018e131f6772c 13.59B
[1]: => 247.4s
[1]: => extracting sha256:1cf9f800d8a24931f7dd9ec730891ee49e112a179bbdbd216989ae1830f775 13.59B
[1]: => 247.4s
[1]: => sha256:1cf9f800d8a24931f7dd9ec730891ee49e112a179bbdbd216989ae1830f775 498B
[1]: => 498B
[1]: => sha256:1cf9f800d8a24931f7dd9ec730891ee49e112a179bbdbd216989ae1830f775 20.88B
[1]: => 20.88B
[1]: => 257.7s
[1]: => extracting sha256:10778025f313aa04316e050801a397160fb718633d5498018e131f6772c
[1]: => sha256:15af14997eaed2255a87875613663968a9794780d19590e9c8589fc6d8e0d1 2.44B
[1]: => 2.44B
[1]: => sha256:15af14997eaed2255a87875613663968a9794780d19590e9c8589fc6d8e0d1 249.1s
[1]: => extracting sha256:1cf9f800d8a24931f7dd9ec730891ee49e112a179bbdbd216989ae1830f775 249.1s
[1]: => 249.1s
[1]: => sha256:15af14997eaed2255a87875613663968a9794780d19590e9c8589fc6d8e0d1 20.83k
[1]: => 20.83k
[1]: => sha256:17865b5d132769f419b544672095859770d7af7a82215ad46767fcf8f09350 31.99B
[1]: => 31.99B
```

The screenshot shows a VMware Workstation interface with an Ubuntu 64-bit VM running. Inside the VM, a browser window is open to a Visual Studio Code terminal. The terminal is titled "dockerfile - Cloud_Computing_Project - Visual Studio Code". The output of the terminal shows the following Docker commands and their results:

```
Jan 11 19:50
dockerfile - Cloud_Computing_Project - Visual Studio Code

14
15 # Copy the application code
16 COPY . .
17
18 # Install Composer
19 COPY --from=composer:latest /usr/bin/composer /usr/bin/composer
20

[+] Network: cloud_computing_project_laravel_network Created
[+] Container: laravel app Started

[{"Name": "laravel_network",
 "Id": "c7ced6d02399d997141184004d073f83e31be15befea3db01591d84922f6",
 "Created": "2023-01-11T17:15:36.4529938+03:30",
 "Scope": "local",
 "Driver": "bridge",
 "EnableIPv6": false,
 "IPAM": {
 "Driver": "default",
 "Options": {},
 "Config": [
 {
 "Subnet": "172.18.0.0/16",
 "Gateway": "172.18.0.1"
 }
 ]
 },
 "Internal": false,
 "Attachable": false,
 "Ingress": false,
 "ConfigFrom": {},
 "Network": "",
 "ConfigOnly": false,
 "Containers": {},
 "Options": {},
 "Labels": {}}
```

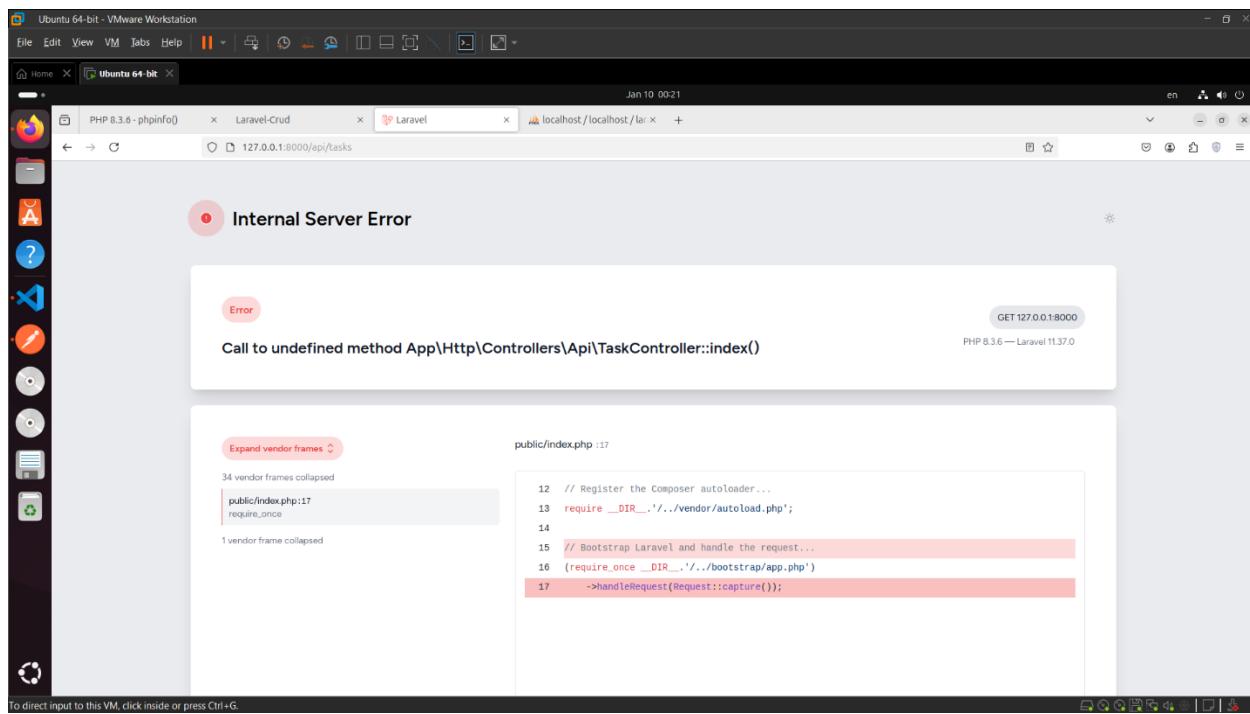
```
FROM php:7.4-fpm
WORKDIR /var/www/html
COPY . /var/www/html
RUN composer install --no-dev --optimize-autoloader
EXPOSE 8000
CMD ["php-fpm"]

# Create laravel directory
RUN curl -sS https://getcomposer.org/installer | php &> composer.phar
RUN mv composer.phar /usr/local/bin/composer
RUN composer config -g repo.packagist composer https://packagist.phpcomposer.com
RUN composer install --no-dev --optimize-autoloader
RUN composer dump-autoload
```

```
Ubuntu-VMware-Platform:~/Cloud_Computing_Project$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                 NAMES
1bacc91cd1    cloud_computing_project-app   "docker-php-entrypoint"   24 seconds ago   Up 21 seconds          0.0.0.0:8080->80/tcp, ::1:8000:80/tcp   laravel_app
Ubuntu-VMware-Platform:~/Cloud_Computing_Project$ sudo docker network inspect laravel_network
```

```
[{"Name": "laravel_network", "Id": "e7cfd9999d0f141184004d73f83a31d158efaf3b01591d84922fd", "Created": "2025-01-11T17:13:38.452959185+01:30", "Scope": "local", "Driver": "bridge", "EnableIPv6": false, "IPAM": { "Driver": "default", "Options": {} }, "Config": [ { "Subnet": "172.18.0.0/16", "Gateway": "172.18.0.1" } ], "Internal": false, "Attachable": false, "Ingress": false, "ConfigFrom": { "Network": "" } }
```

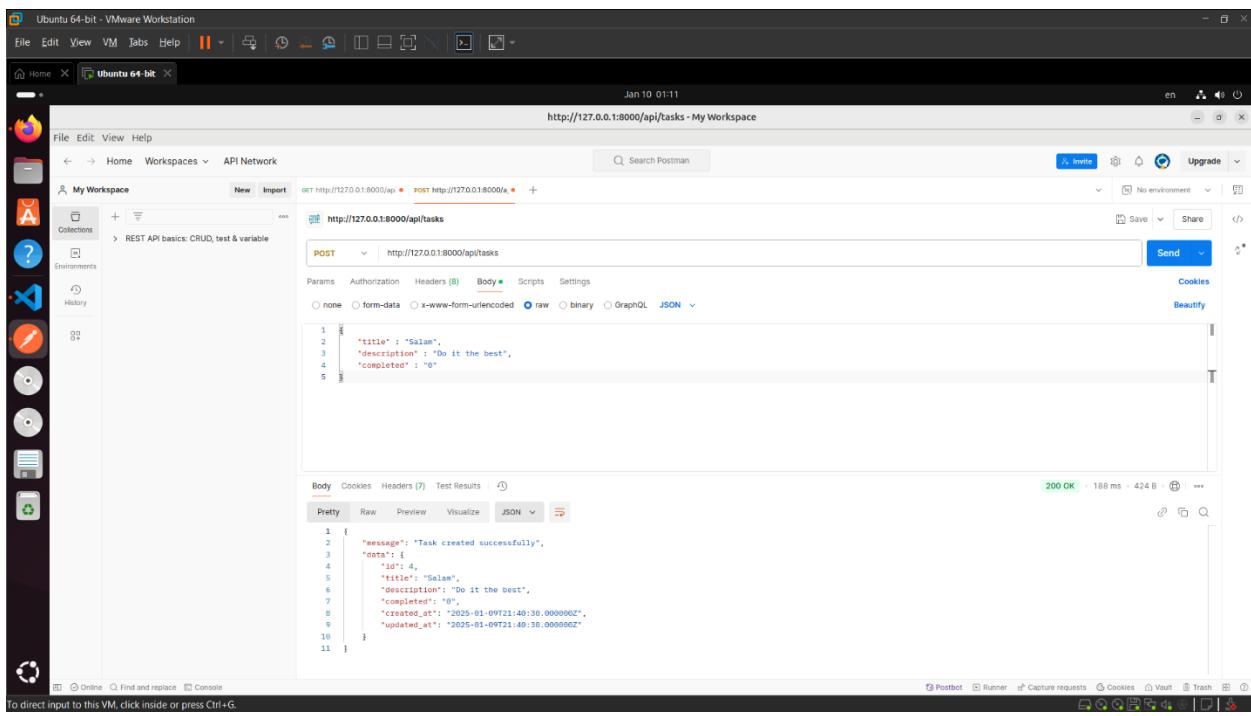

تست دیتاها با POSTMAN

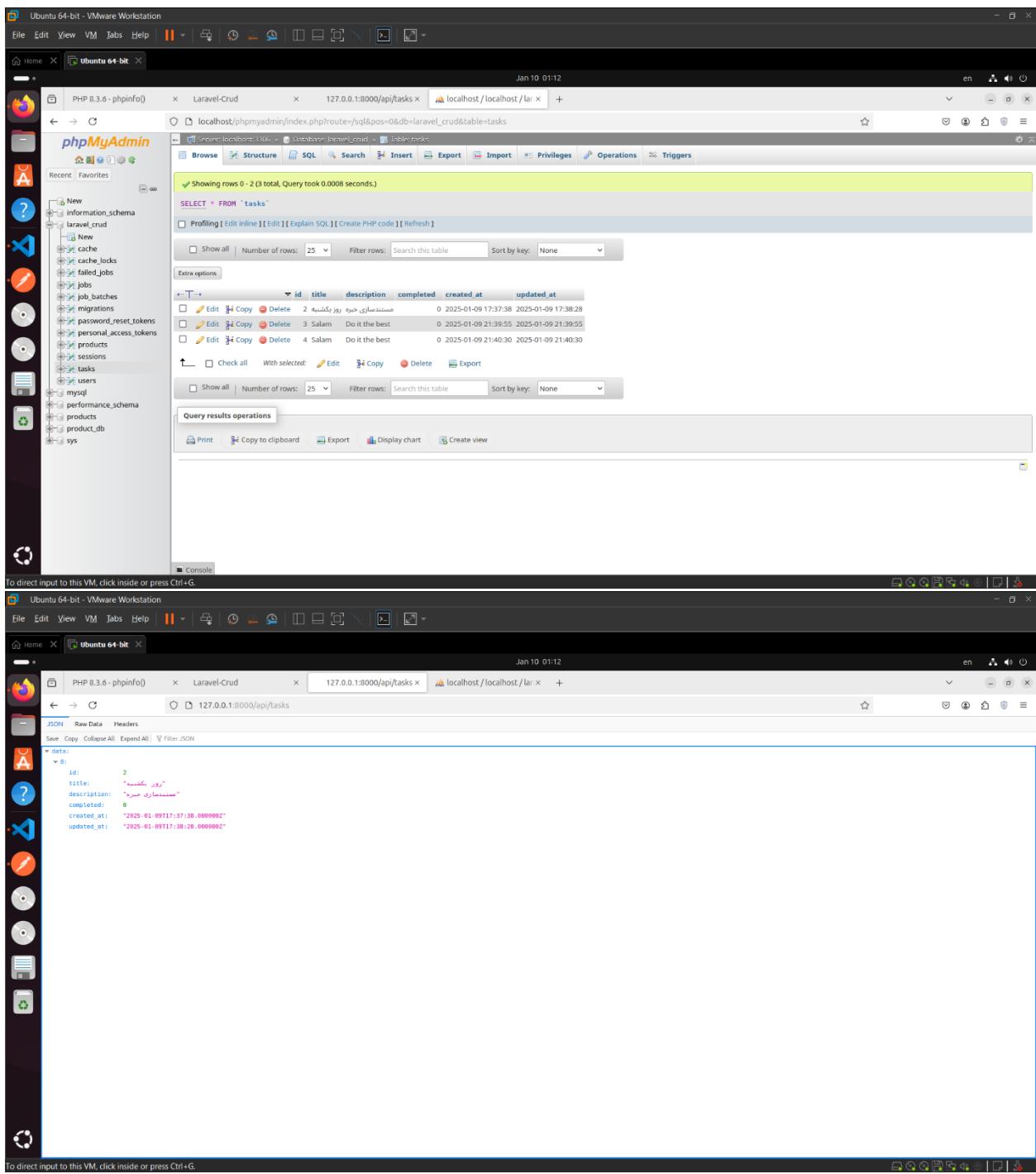


The screenshot shows the Postman application interface. A search bar at the top has 'http://127.0.0.1:8000/api/tasks - My Workspace' entered. Below the search bar, there's a list of collections: 'My Workspace', 'Collections', 'Environments', and 'History'. A single item 'REST API basics: CRUD, test & variable' is listed under 'My Workspace'. In the center, there's a request card for a 'GET' request to 'http://127.0.0.1:8000/api/tasks'. The 'Params' tab is selected, showing a table with one row: 'Key' (id) and 'Value' (3). The 'Body' tab is selected, showing a JSON response:

```
1 {
2   "data": [
3     {
4       "id": 2,
5       "title": "aaaaa",
6       "description": "aaaaa",
7       "completed": 0,
8       "created_at": "2025-01-09T17:37:38.000000Z",
9       "updated_at": "2025-01-09T17:38:28.000000Z"
10    },
11    {
12      "id": 3,
13      "title": "Salam"
14    }
15  ]
16}
```

The status bar at the bottom indicates '200 OK' with a response time of '215 ms'.





The screenshot shows the Postman application interface on an Ubuntu 64-bit VM. A PUT request is being made to `http://127.0.0.1:8000/api/tasks/3`. The request body contains the following JSON:

```
1 {  
2   "title": "Sala",  
3   "description": "Do",  
4   "completed": 1  
5 }
```

The response status is 200 OK, with a response time of 140 ms and a response size of 409 B. The response body is:

```
1 {  
2   "message": "Task updated successfully",  
3   "data": {  
4     "id": 3,  
5     "title": "Sala",  
6     "description": "Do",  
7     "completed": 1,  
8     "created_at": "2025-01-09T21:39:55.000000Z",  
9     "updated_at": "2025-01-09T22:14:45.000000Z"  
10   }  
11 }
```


The screenshot shows the Postman application interface on an Ubuntu 64-bit VM. A DELETE request is being made to `http://127.0.0.1:8000/api/tasks/3`. The response status is 200 OK, with a response time of 169 ms and a response size of 259 B. The response body is:

```
1 {  
2   "message": "Task deleted successfully"  
3 }
```

The screenshot shows two identical instances of the Postman application running on an Ubuntu 64-bit VM. Both instances are displaying a GET request to `http://127.0.0.1:8000/api/tasks/3`. The requests are failing with a 404 Not Found status code. The response body shows a standard HTML error page from the Normalize.css project.

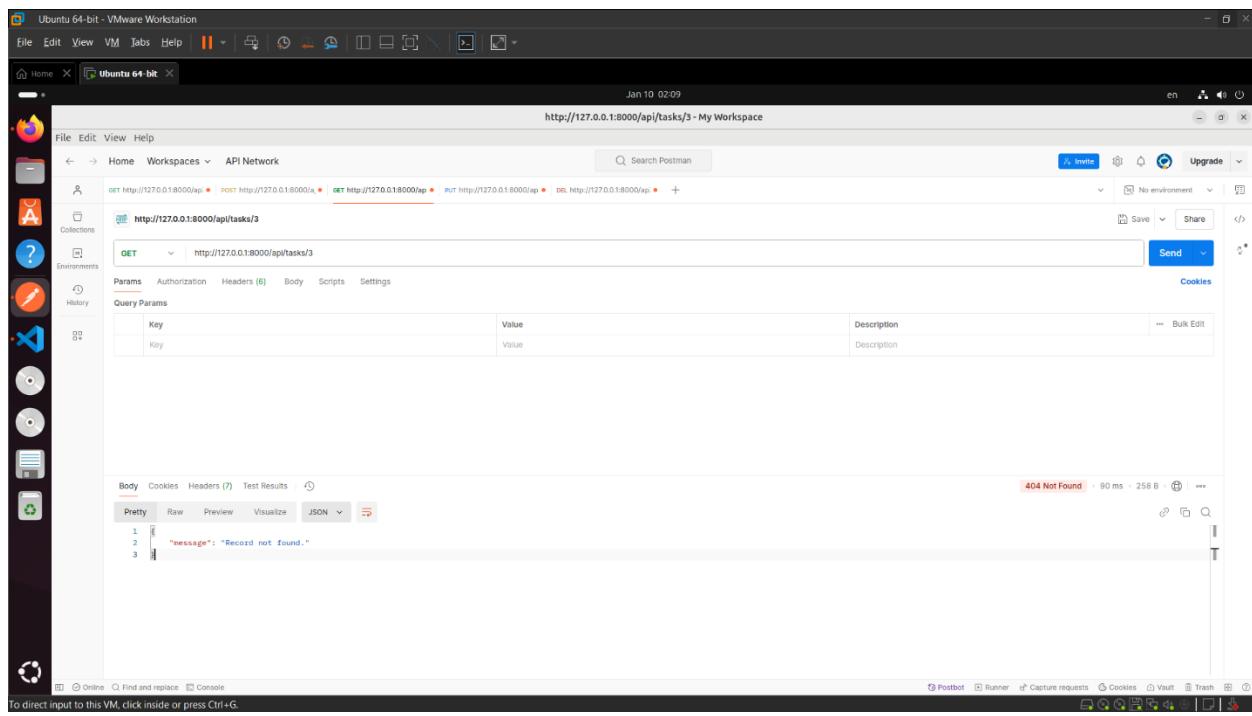
Request URL: http://127.0.0.1:8000/api/tasks/3

Status: 404 Not Found

Body:

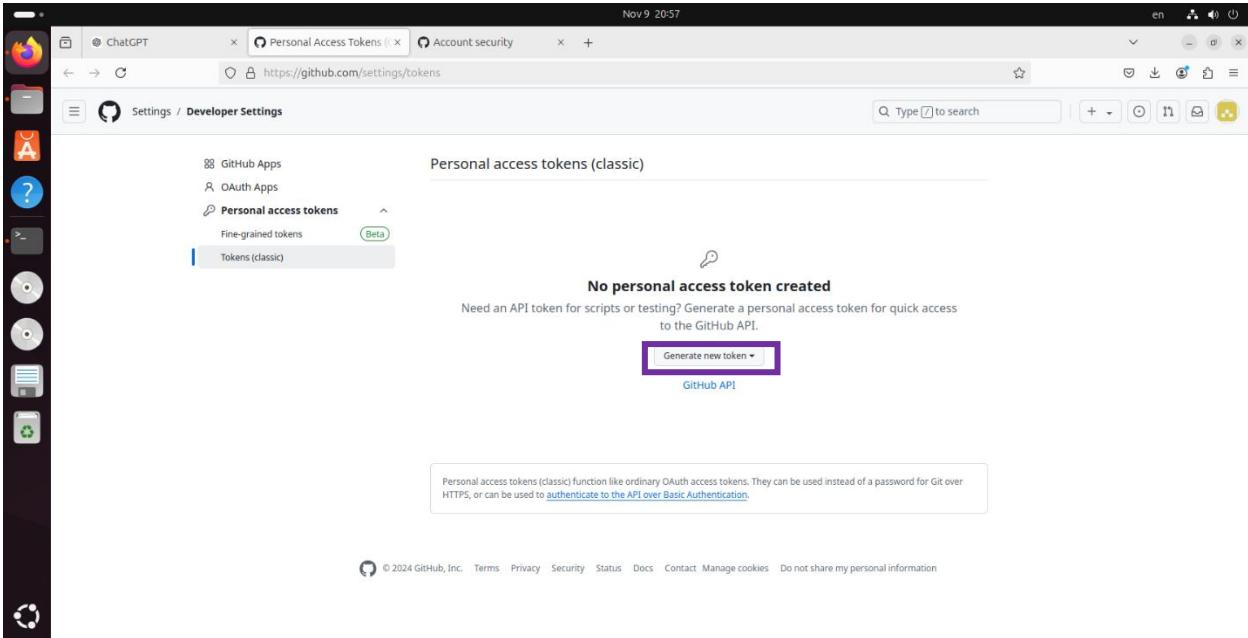
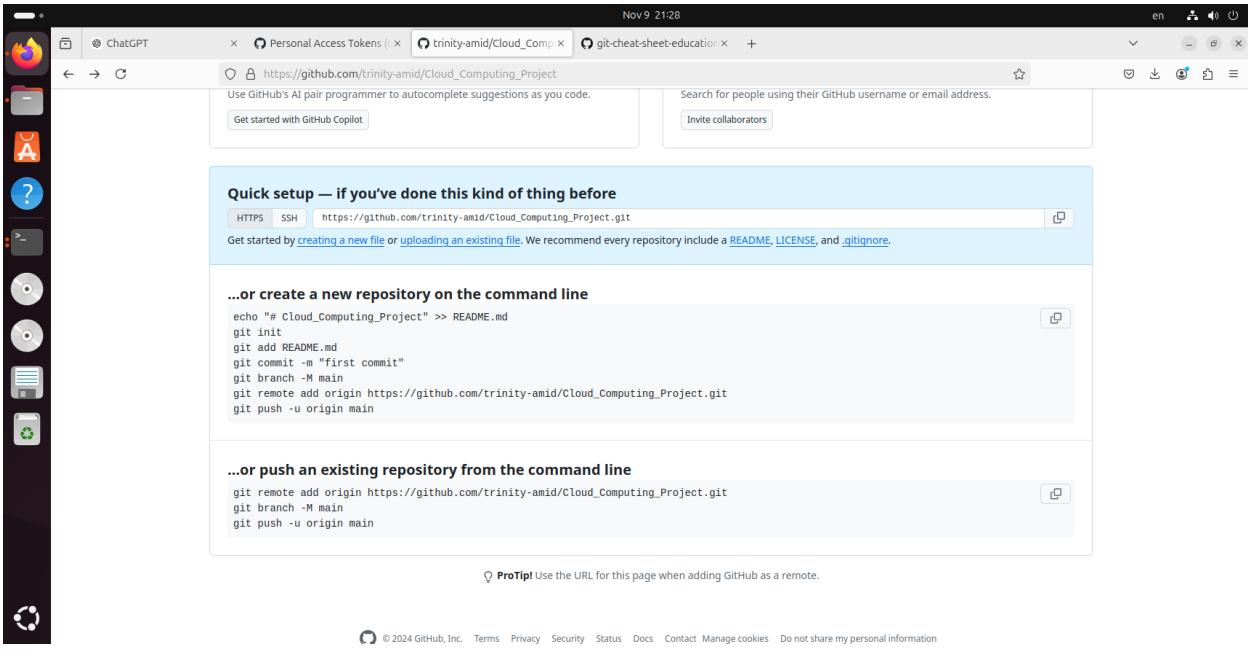
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Not Found</title>
    <style>
        /*! normalize.css v0.0.1 | MIT License | github.com/necolas/normalize.css */
        html {
            line-height: 1.15;
        }
    </style>

```



ایلوود داکیومنت پیروزه روی گیتهاب:

The screenshot shows the GitHub Home page. It features a search bar at the top right, a dashboard icon, and a 'Dashboard' button. Below the dashboard, there's a 'Top repositories' section with a 'New' button and a search bar. The main area has sections for creating a new repository ('Start writing code'), introducing自己 with a profile README, and viewing the 'UNIVERSE'24' section. On the right, there's a 'Latest changes' sidebar listing recent updates.



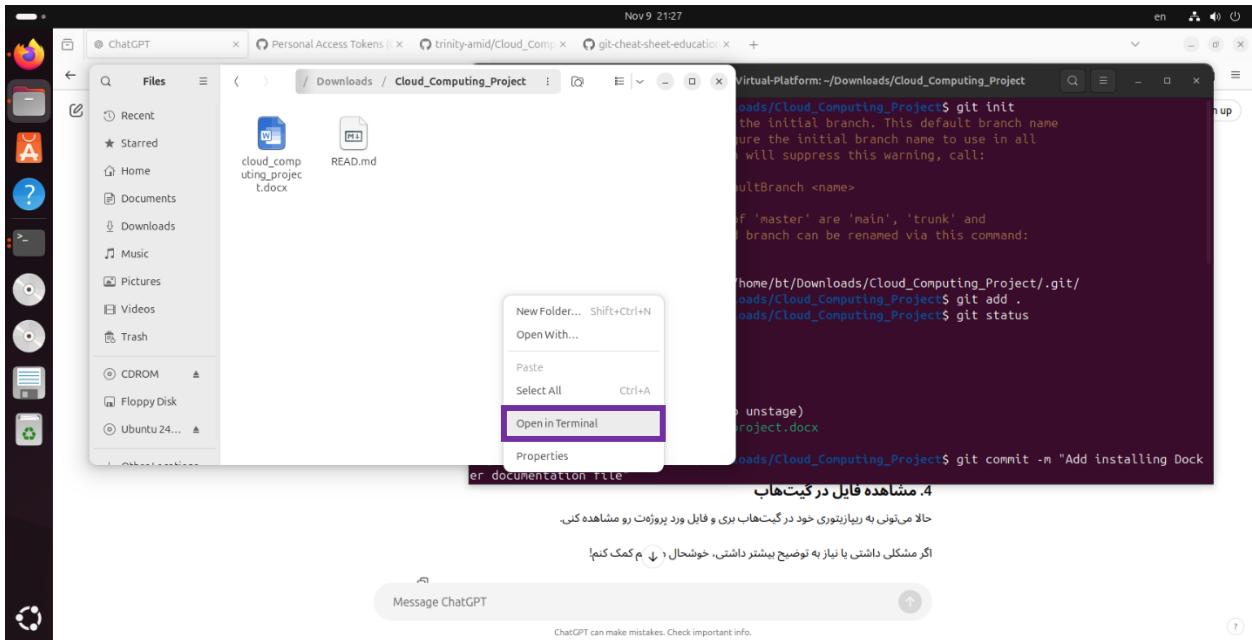
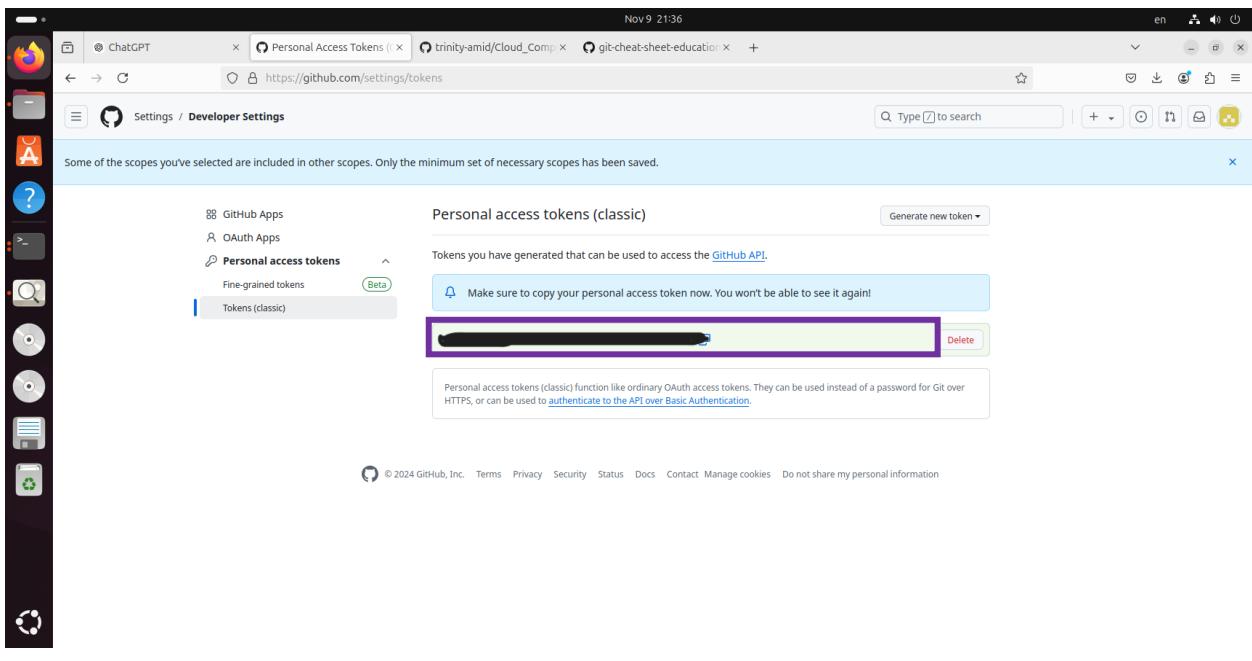
The screenshot shows the GitHub settings interface for creating a new personal access token. The URL is <https://github.com/settings/tokens>. The page title is "New personal access token (classic)". A note at the top states: "Some of the scopes you selected are included in other scopes. Only the minimum set of necessary scopes has been saved." Below this, a red error message box says "Note can't be blank". The "Personal access tokens" section is selected, showing "Tokens (classic)" as the current tab. A note below the tabs says "Beta". The main form fields include:

- Note:** cloud_computing_project (highlighted with a purple box)
- Expiration:** 30 days (highlighted with a purple box)
- Select scopes:** A list of GitHub scopes:
 - repo** (selected):
 - manage_runners:enterprise
 - manage_billing:enterprise
 - read:enterprise
 - scim:enterprise
 - audit_log**:
 - manage_audit_log
 - read:audit_log
 - codespace**:
 - codespace:secrets
 - copilot**:
 - manage_billing:copilot
 - project**:
 - manage_projects
 - read:project
 - admin:gpg_key**:
 - manage_gpg_keys
 - write:gpg_key
 - read:gpg_key
 - admin:ssh_signing_key**:
 - manage_ssh_signing_keys
 - write:ssh_signing_key
 - read:ssh_signing_key
- workflow** (selected):
 - update_github_action_workflows

The screenshot shows the GitHub settings interface for creating a new personal access token. The URL is <https://github.com/settings/tokens/new>. The page title is "New Personal Access Token". The main content area displays a large table of GitHub scopes:

<input type="checkbox"/> manage_runners:enterprise	Manage enterprise runners and runner groups
<input type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input type="checkbox"/> scim:enterprise	Provisioning of users and groups via SCIM
<input type="checkbox"/> audit_log	Full control of audit log
<input type="checkbox"/> read:audit_log	Read access of audit log
<input type="checkbox"/> codespace	Full control of codespaces
<input type="checkbox"/> codespace:secrets	Ability to create, read, update, and delete codespace secrets
<input type="checkbox"/> copilot	Full control of GitHub Copilot settings and seat assignments
<input type="checkbox"/> manage_billing:copilot	View and edit Copilot Business seat assignments
<input type="checkbox"/> project	Full control of projects
<input type="checkbox"/> manage_projects	Read access of projects
<input type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input type="checkbox"/> manage_gpg_keys	Write public user GPG keys
<input type="checkbox"/> read:gpg_key	Read public user GPG keys
<input type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> manage_ssh_signing_keys	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

At the bottom of the table are two buttons: "Generate token" (highlighted with a purple box) and "Cancel".



Nov 9 21:27

```
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/bt/Downloads/Cloud_Computing_Project/.git/
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git add .
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   cloud_computing_project.docx

bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git commit -m "Add installing Docker documentation file"
4. مشاهده فایل در گیت هاب
```

حالا من توانم به ریپوزیتوری خود در گیت هاب برو و فایل ورد پرورزه داشته باشم.

اگر مشکل داشتم با نیاز به توضیح بیشتر داشتم، خوشحال باشم کمک کنم!

Message ChatGPT

ChatGPT can make mistakes. Check important info.

Nov 9 21:28

```
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git commit -m "Add installing Docker documentation file"
[master (root-commit) ae068a8] Add installing Docker documentation file
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git remote add origin https://github.com/trinity-amid/Cloud_Computing_Project.git
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ echo "#Cloud_Computing_Project" >> README.md
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ echo "#Cloud_Computing_Project" >> README.md
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git add .
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   README.md

bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git commit -m "Add installing Docker documentation"
[master 38da5e3] Add installing Docker documentation
  1 file changed, 2 insertions(+)
  create mode 100644 README.md
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git branch -M main
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git remote add origin https://github.com/trinity-amid/Cloud Computing Project.git
4. مشاهده فایل در گیت هاب
```

حالا من توانم به ریپوزیتوری خود در گیت هاب برو و فایل ورد پرورزه داشته باشم.

اگر مشکل داشتم با نیاز به توضیح بیشتر داشتم، خوشحال باشم کمک کنم!

Message ChatGPT

ChatGPT can make mistakes. Check important info.

The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open in the background, displaying a command-line session for committing changes to a GitHub repository. The session includes creating a branch, adding files, and pushing them to the remote origin.

```
Nov 9 21:28
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git commit -m "Add installing Docker documentation"
[master 38da5e3] Add installing Docker documentation
 1 file changed, 2 insertions(+)
 create mode 100644 README.md
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git branch -M main
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git remote add origin https://github.com/trinity-amid/Cloud_Computing_Project.git
error: remote origin already exists.
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$ git push -u origin main
Username for 'https://github.com': trinity-amid
Password for 'https://trinity-amid@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 11.12 MiB | 633.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/trinity-amid/Cloud_Computing_Project.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
bt@bt-VMware-Virtual-Platform:~/Downloads/Cloud_Computing_Project$
```

Message ChatGPT

حال چون نه به ریپازنور خود در گیت‌هاب برو و فایل ورد پروژت رو مشاهده کنی.
اگر مشکل داشتی با نیاز به توضیح بیشتر داشتی، خوشحال باشم کمک کنم!

Nov 9 21:28

trinity-amid / Cloud_Computing_Project

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Cloud_Computing_Project (Public)

main 1 Branch 0 Tags Go to file Add file Code

trinity-amid Add installing Docker documentation 38da5e3 · 2 minutes ago 2 Commits

README.md Add installing Docker documentation 2 minutes ago

cloud_computing_project.docx Add installing Docker documentation file 6 minutes ago

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

About

No description, website, or topics provided.

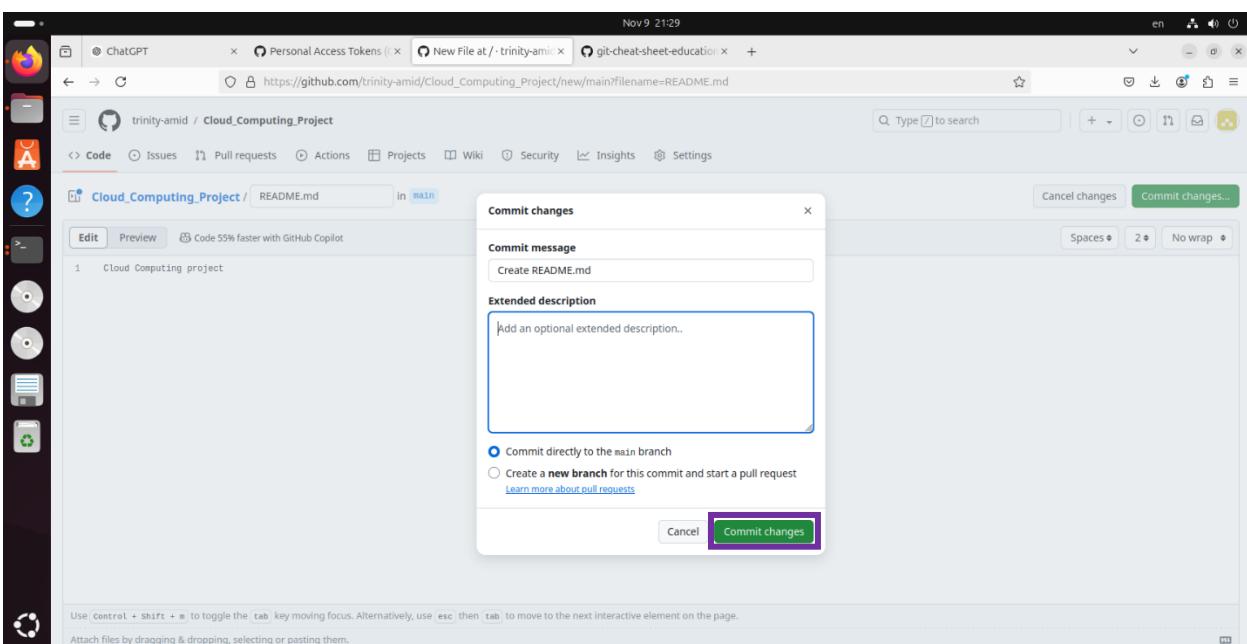
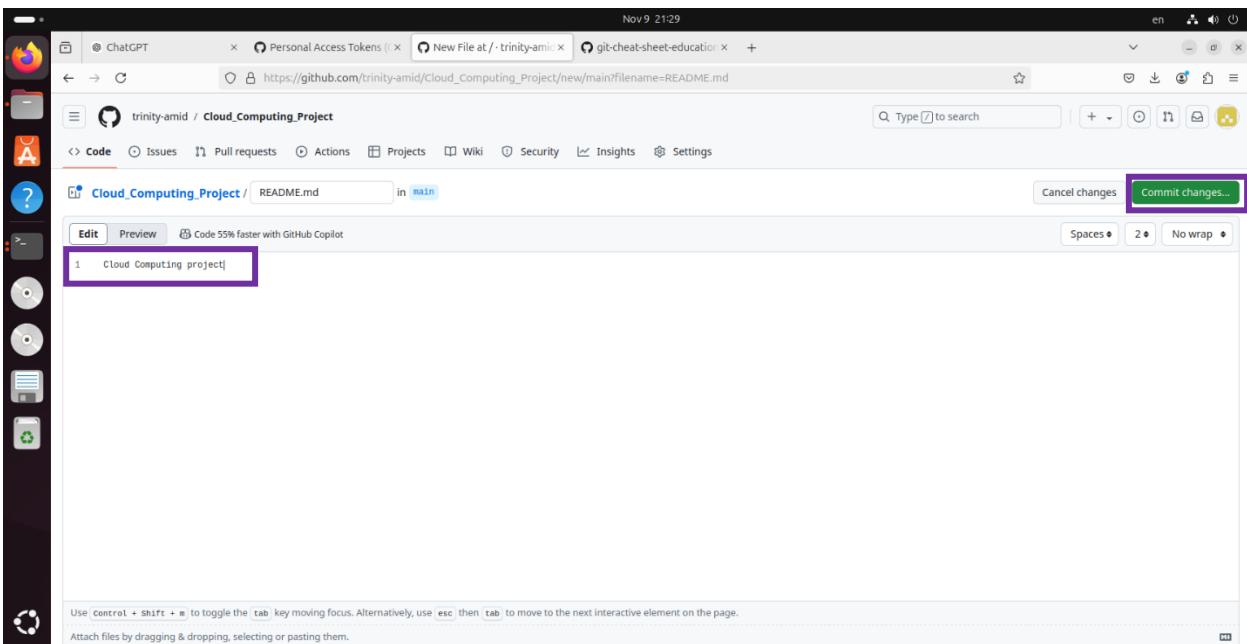
Activity 0 stars 1 watching 0 forks

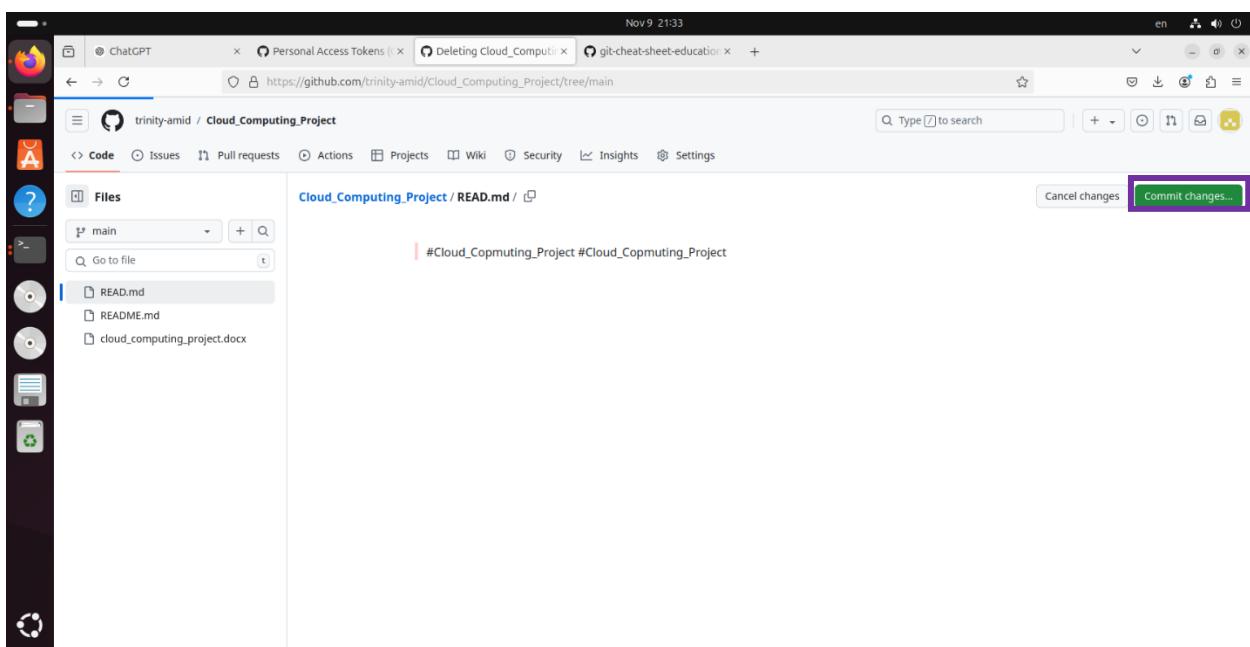
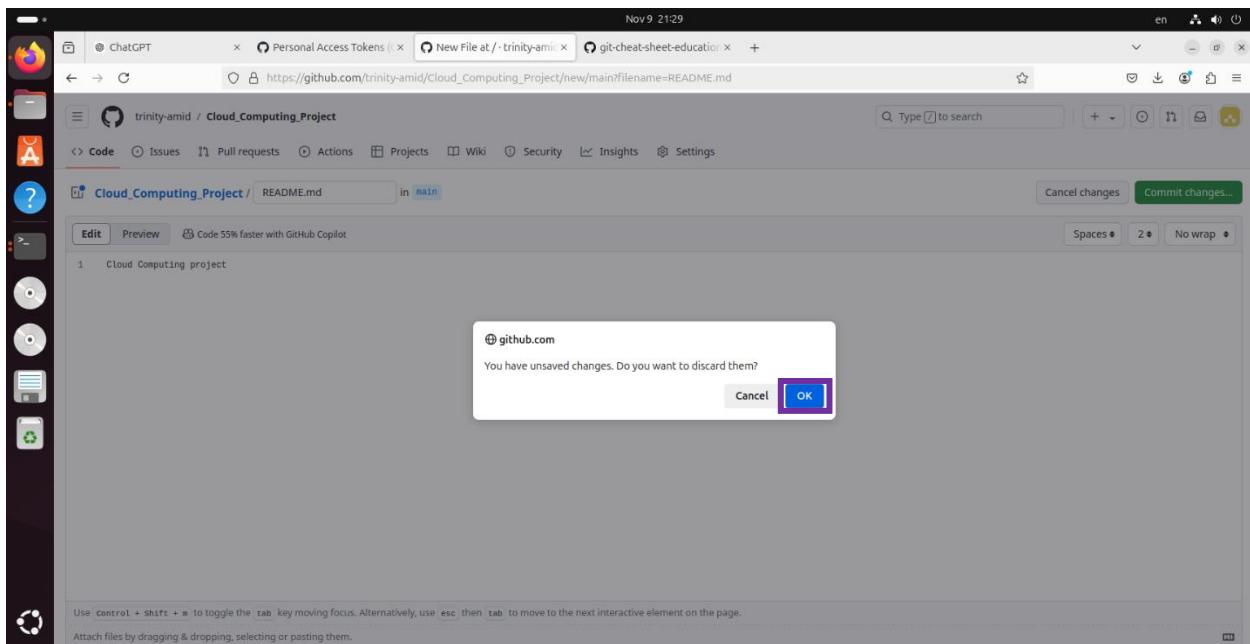
Releases

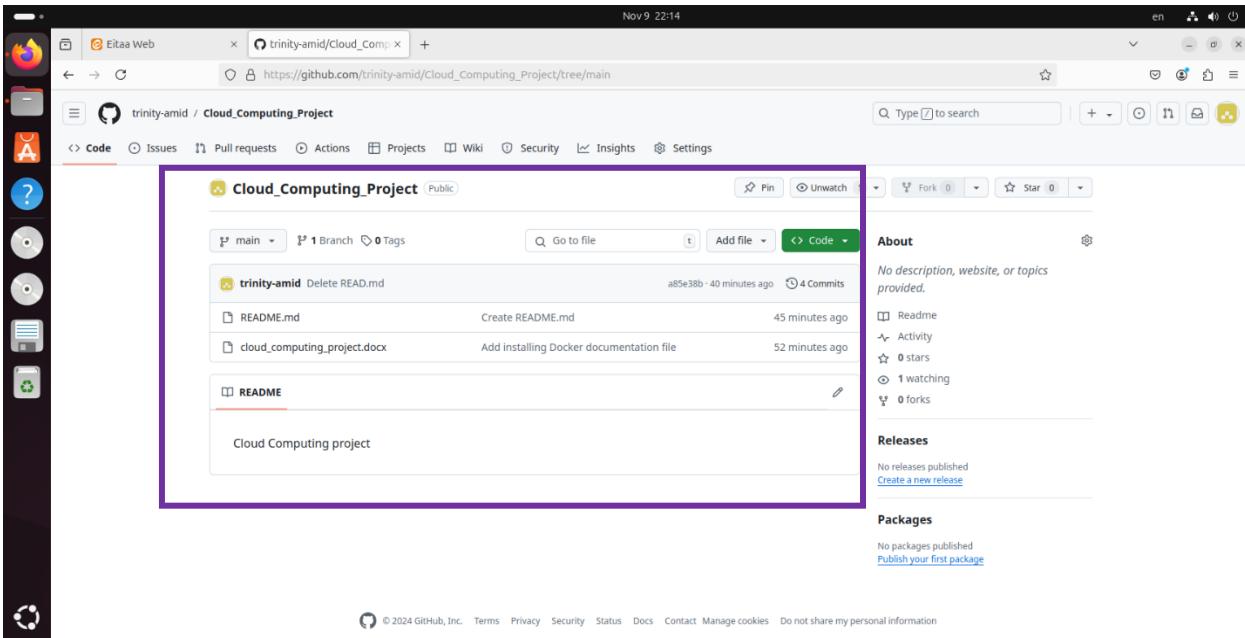
No releases published Create a new release

Packages

No packages published Publish your first package







SSH (Secure Shell) و HTTPS (Hypertext Transfer Protocol Secure) دو پروتکل مختلف برای ارتباط امن با سرورها و مخازن گیت هستند. هر کدام کاربردها و ویژگی‌های خاص خود را دارند:

SSH:

- رمزنگاری: اطلاعاتی که از طریق SSH منتقل می‌شود، به صورت رمزنگاری شده است که امنیت بالایی را فراهم می‌کند.
- احراز هویت: برای استفاده از SSH، معمولاً از کلیدهای عمومی و خصوصی برای احراز هویت استفاده می‌شود. این به معنای این است که نیاز نیست هر بار نام کاربری و رمز عبور را وارد کنید.
- استفاده: بیشتر برای توسعه‌دهندگان و برنامه‌نویسانی که به سرورهای دوردست دسترسی دارند، مناسب است.

HTTPS:

- پروتکل وب HTTPS: یک نسخه امن از HTTP است که برای انتقال اطلاعات بین مرورگر و وبسایت‌ها استفاده می‌شود.
 - آسانی استفاده HTTPS: معمولاً برای کاربران معمولی که به مخازن گیت مراجعه می‌کنند، مناسب‌تر است و نیازی به تنظیم کلیدهای SSH ندارد. فقط کافی است نام کاربری و رمز عبور خود را وارد کنید.
 - گسترش کاربرد: بهطور گسترده در وبسایت‌ها و اپلیکیشن‌ها استفاده می‌شود و امنیت انتقال داده‌ها را تضمین می‌کند.
- در نهایت، انتخاب بین SSH و HTTPS بستگی به نیازهای امنیتی و راحتی کاربر دارد. برای بیشتر کارهای توسعه، SSH به دلیل امنیت بیشتر توصیه می‌شود.

README.me ذخیره شده **markdown** یا پسوند **.md** هست

References:

- <https://www.docker.com/get-started/>
- <https://www.youtube.com/@BobyCloud>
- <https://www.ibm.com/topics/docker>
- <https://dockerme.ir/>
- <https://youtu.be/WumgBzENYYk?si=B9wLRIZ15sH7MNZN>
- https://youtube.com/playlist?list=PLwu_zull7CDN6x_VokJnlv63AMCcxW79&si=KPiD1MnJnF5ga_Bi
- ChatGPT.com