

باسمه تعالی



دانشگاه آزاد اسلامی
واحد اصفهان (خوراسگان)
دانشکده فنی و مهندسی

پایان نامه برای دریافت درجه کارشناسی در رشته مهندسی کامپیوتر
گرایش نرم افزار

عنوان

روش نشر بهینه سایت در فضای مجازی

استاد راهنما

دکتر ثقفی نیا

نگارنده

امین جمالی

بهمن 1403

پروردگار ایاری کن تا دانش اندکم نه زربانی باشد برای فزونی و تکبر و غرور، نه حشای برای اسارت و نه دستبندی برای تجارت، بلکه گامی باشد برای تحلیل از تو و تعالی ساختن زندگی خود و دیگران.

قبل از هر چیزی، خداوند بزرگ را به خاطر لطفی که بهواره شامل حال من نموده شاکرم. پس، از زحمات استاد محترم را بنما، جناب آقای ثقفی نیا که نه تنها به عنوان استاد بلکه به چون بهکاری در تمام مراحل انجام این تحقیق از رهنمود و کمک های بی دریغ ایشان بهره مند شده ام، مشکور و قدردانی می کنم.

چکیده

این پروژه با عنوان مقرون به صرفه ترین روش نشر سایت ، پروژه‌ای است که با بهترین و بروز ترین نرم-افزارهای رایج طراحی و آماده شده است. هدف از ایجاد این پروژه پیدا کردن بروز ترین و مقرون به صرفه ترین روش برای طراحی و انتشار سایت در فضای اینترنت است.

فهرست مطالب

عنوان	صفحه
1- فصل اول کلیات پروژه.....	1
1-1- بیان کلیات مسئله.....	1
1-2- نیازمندی های پروژه.....	1
1-3- نرم افزارهای استفاده شده در پیاده سازی سیستم.....	2
2- فصل دوم طراحی سیستم.....	3
2-1- مقدمه.....	3
2-2- نرم افزار کد نویسی و اتصال SSH.....	3
2-3- دامنه (Domain).....	3
2-4- هاست (host).....	9
3- فصل سوم پیاده سازی سیستم.....	15
3-1- مقدمه.....	15
3-2- نصب داکر (Docker).....	15
3-3- فایل داکر کامپوس (Docker Compose).....	17
3-3-1- سرویس WordPress.....	18
4-1-1- سرویس دیتابیس (wordpress_db).....	20
4-1-2- سرویس کشینگ ردیس.....	21
4-1-3- شبکه ها (Networks).....	21
4-2- نصب nginx.....	22
4-3- نصب certbot.....	23
4-4- گرفتن توکن کلودفلر.....	25
4-4-1- انتشار سایت با SSL به وسیله nginx و certbot.....	31
4-4-2- تنظیم NGINX.....	32
4-5- فواید اجرای سایت روی سرور به جای خرید سرور هاستینگ.....	34
4-5-1- هزینه.....	34
4-5-2- حریم خصوصی.....	35
4-5-3- به روز بودن.....	36
4-5-4- افزودن قابلیت به پروژه با نرم افزار های متن باز.....	36
4-5-5- استفاده از ظرفیت آزاد سرور.....	36
4-5-6- افزایش امتیاز عملکرد سایت وردپرس.....	36

364-5-7 محدود بودن به وردپرس
375- فصل چهارم جمع بندی
375-1- جمع بندی مراحل قبل
375-2- نتیجه گیری و پیشنهادات
375-3- منابع و مراجع

فهرست شکل ها

4	تصویر 2-1: سایت نیم چیپ
5	تصویر 2-2 جستجو در سایت نیم چیپ
5	تصویر 2-3 جستجو در نیم چیپ
6	تصویر 2-4 ثبت نام در سایت کلودفلر
6	تصویر 2-5 وارد شدن به حساب کلودفلر
7	تصویر 2-6 اضافه کردن دامنه
7	تصویر 2-7 اضافه کردن دامنه
8	تصویر 2-8 انتخاب پلن کلودفلر
8	تصویر 2-9 نیم سرورهای کلودفلر
8	تصویر 2-10 تنظیمات دامنه در نیم چیپ
10	تصویر 2-11 صفحه اصلی ویژوال استودیو کد
10	تصویر 2-12 باز کردن ترمینال جدید
11	تصویر 2-13 نصب افزونه SSH
11	تصویر 2-14 تنظیمات SSH
12	تصویر 2-15 اضافه کردن هاست
ERROR! BOOKMARK NOT DEFINED.	تصویر 2-16 اضافه کردن هاست
13	تصویر 2-17 اتصال به هاست

فصل اول

کلیات پروژه

1-1- بیان کلیات مسئله

آنچه شما پیش‌رو دارید پروژه‌ای است برای طراحی و نشر سایت که در این بخش به توجیه و توصیف این موضوع پرداخته خواهد شد.

هدف از طراحی و پیاده‌سازی این سیستم، روش و تکنولوژی‌های جدید برای طراحی و نشر یک وب سایت پر سرعت است در حالی که هزینه این کار را به حداقل برساند.

این سیستم کلیه عملیات را که عمدتاً با پرداخت هزینه و به شکل غیر مطلوب انجام میشود را بدون هزینه و به بهترین و بروز ترین شکل انجام دهد.

1-2- نیازمندی های پروژه

برای انجام این پروژه ابتدا نیازها و ابزارهای طراحی و انتشار سایت بررسی شد و نیازمندی های زیر حاصل شد:

1. پلتفرم طراحی سایت

طراحی سایت میتواند بدون هیچ پلتفرمی انجام شود اما استفاده از پلتفرم باعث آسان تر شدن کار و افزایش سرعت کار می شود برخی از این پلتفرم ها عبارتند از:

وردپرس: معروف ترین پلتفرم طراحی سایت که حدود سی درصد کل سایت های اینترنتی توسط این پلتفرم بنا شده و یک پلتفرم ساده برای طراحی سایت های مختلف بدون نیاز به دانش کد نویسی است.

فلاتر: فریم ورکی از شرکت گوگل که اجازه می دهد با یک کد برنامه برای وب و ویندوز و اندروید و آی او اس طراحی کنیم
ری اکت: پلتفرم با تمرکز بر طراحی سایت و وب اپ که سئو آن بهینه شده
2. دامنه

خرید دامنه متناسب با وب سایت مد نظر ما و گرفتن گواهی اس اس ال و سی دی ان مناسب
3. هاست

ماشین مجازی ابری که وب سایت روی آن میزبانی شود روی پلتفرمی که اجازه کاهش یا افزایش قدرت ماشین با توجه به نیازهای فعلی سایت و تعداد بیننده را داشته باشد
4. فایروال

برای جلوگیری از حملات سایبری و حفاظت از سایت

با توجه به این نیاز ها هدف پیاده سازی یک سیستم با ابزار مناسب است که در آن بتوان یک سایت را به شکل مقرون به صرفه و با کارایی بالا طراحی و منتشر کرد.

3-1- نرم افزارهای استفاده شده در پیاده سازی سیستم

برای ساخت نرم افزار، از نرم افزار Visual studio code برای نوشتن کد و فرمت آن و برای اتصال به ماشین های مجازی استفاده میشود و قابلیت این نرم افزار با نصب افزونه ها (Extensions) افزایش می یابد

فصل دوم

طراحی سیستم

1-2- مقدمه

در این فصل به بررسی پروژه و طراحی سیستم می پردازیم در این فصل موارد مورد نیاز پروژه را شناسایی کرده و در نهایت در فصل 3 این موارد را تامین و برای استفاده در پروژه پیاده سازی میکنیم

2-2- نرم افزار کد نویسی و اتصال SSH

ما برای نوشتن کد و اتصال SSH از نرم افزار متن باز و مجانی ویژوال استودیو کد استفاده میکنیم و برای اضافه کردن قابلیت های مورد نیاز به این نرم افزار در هر مرحله افزونه های مورد نیاز را نصب می کنیم برای نصب این نرم افزار در ویندوز نرم افزار powershell را باز میکنیم و فرمان زیر را اجرا میکنیم

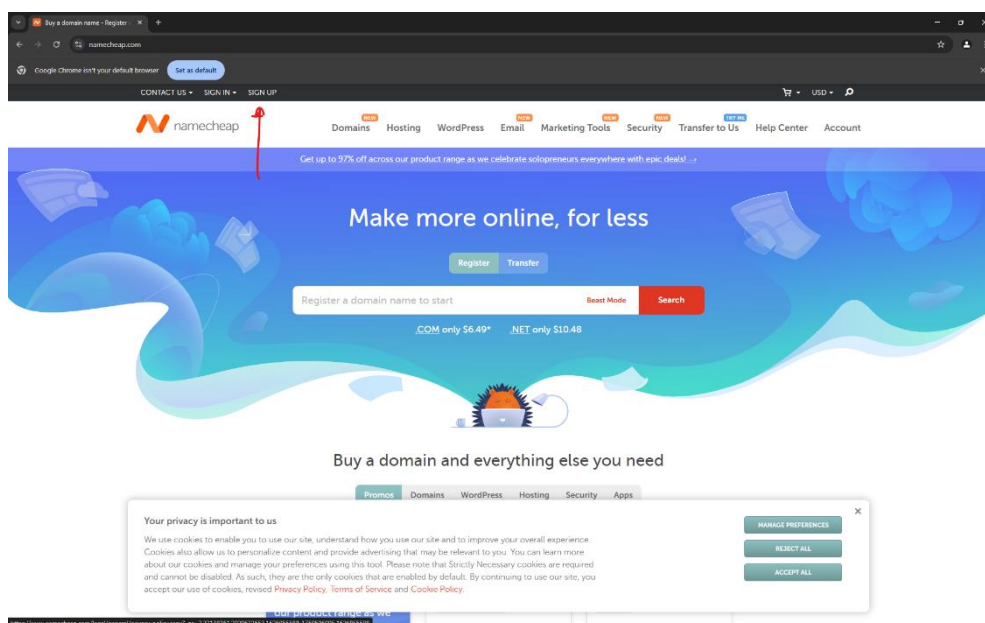
```
winget install --id "Microsoft.VisualStudioCode"
```

3-2- دامنه (Domain)

اولین نیازمندی که به ذهن میرسد خرید یک دامنه است تا کاربران از طریق آن بتوانند به سایت ما دسترسی پیدا کنند دو سایتی که بهترین تعرفه ها را ارائه برای خرید دامنه ارائه میدهند نیم چپ (name cheap) و سایت کلودفلر است (Cloudflare)

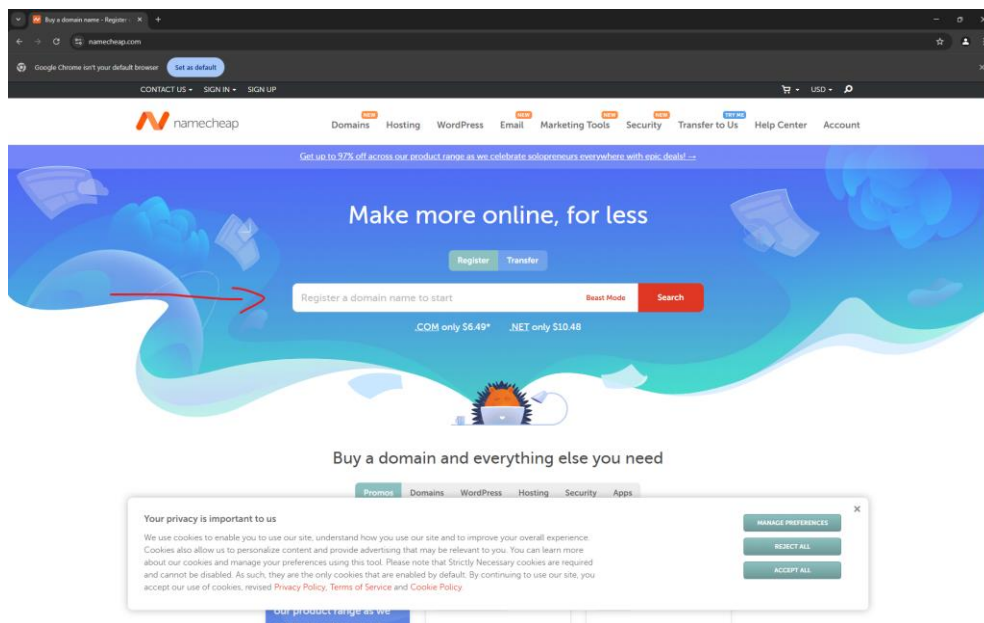
برای خرید از این سایت ها میتوان از سایت هایی مثل ایرانی کارت (iranicard.ir) برای پرداخت هزینه های این سایت ها استفاده کرد

در این سایت ها میتوانیم سایت های به ارزانی یک دلار برای یک سال خریداری کنیم برای این پروژه ما از سایت نیم چیپ دامنه را خریداری کرده و آن را به سایت کلودفلر برای استفاده از سی دی آن (CDN:Content delivery network) استفاده میکنیم ابتدا به سایت نیم چیپ رفته و یک اکانت میسازیم



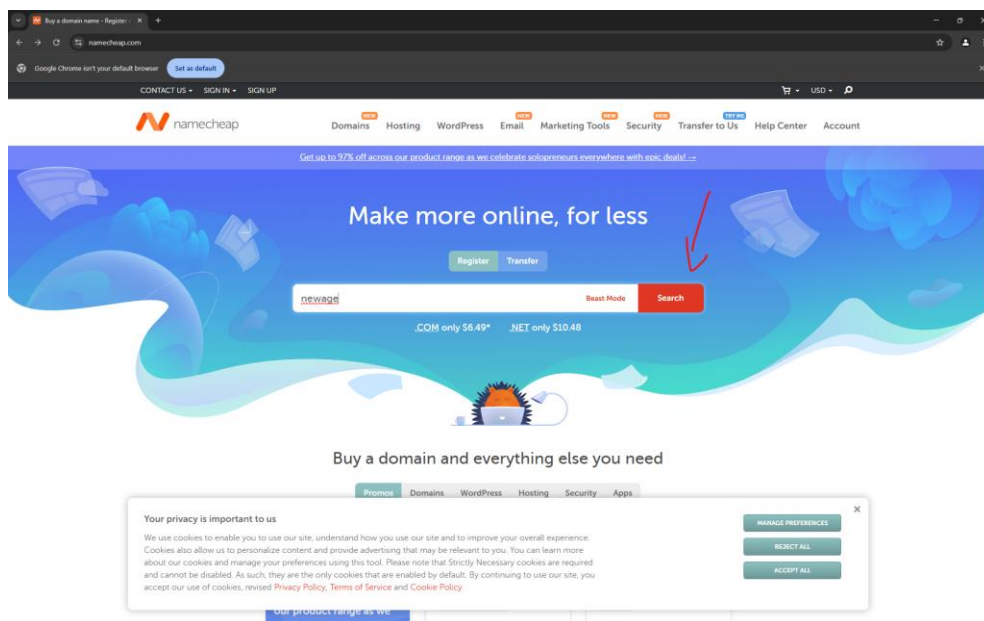
تصویر 1-2: سایت نیم چیپ

سپس دامنه مورد نظر خود را بدون پسوند وارد و از بین گزینه ها دامنه مناسب را انتخاب میکنیم



تصویر 2-2 جستجو در سایت نیم چپ

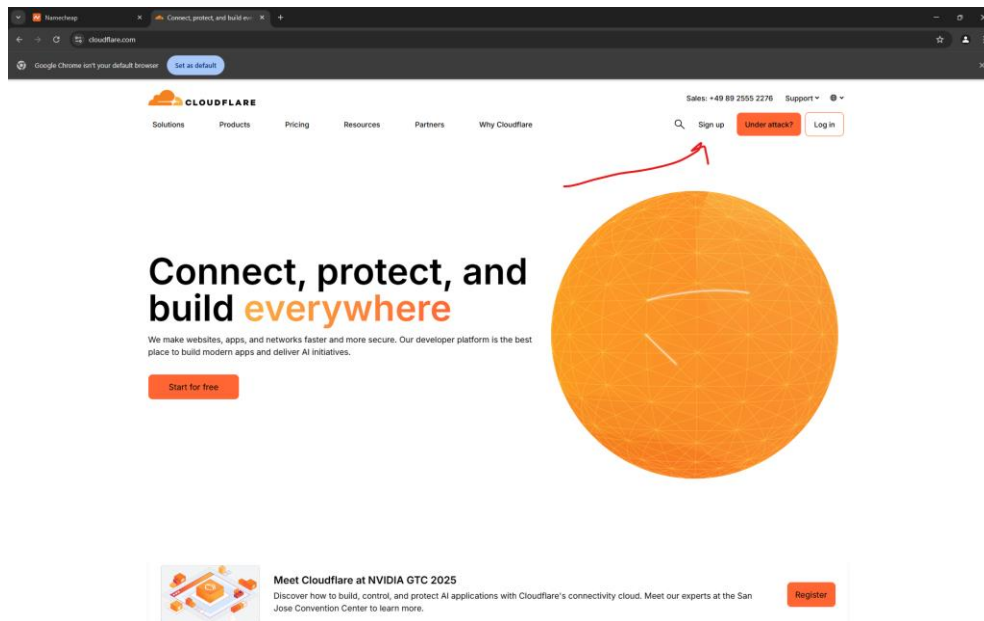
و گزینه جستجو را میزنیم



تصویر 2-3 جستجو در نیم چپ

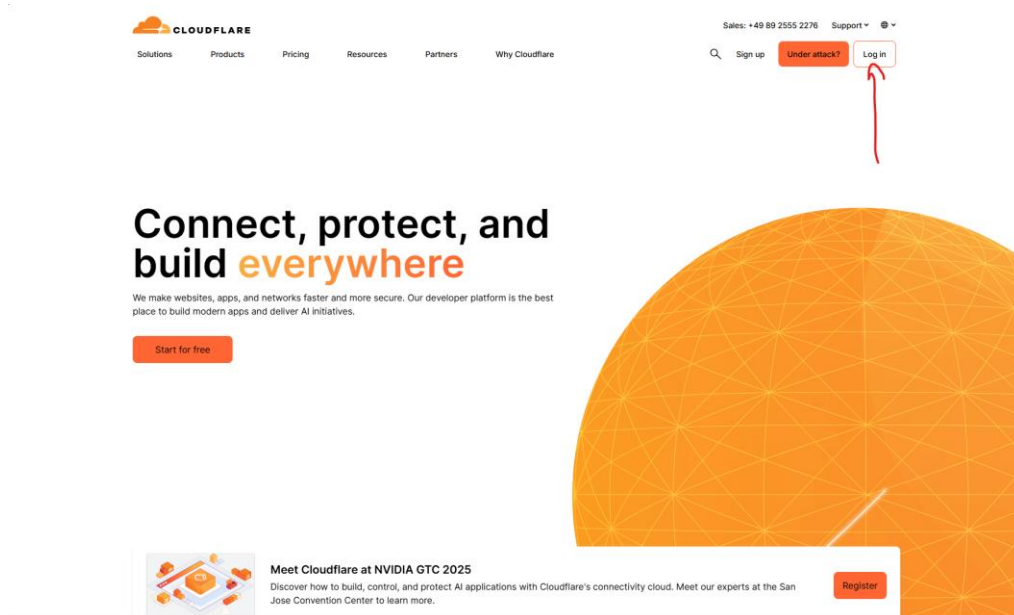
از بین گزینه های ظاهر شده دامنه مناسب را انتخاب و با کمک سایت های پرداخت خارجی مثل ایرانی کارت دامنه مناسب خود را خریداری میکنیم
برای این پروژه دامنه ما newage.rest هست حالا میخواهیم دامنه را به سایت کلودفلر انتقال بدهیم

ابتدا به نشانی cloudflare.com رفته و روی گزینه ثبت نام کلیک میکنیم



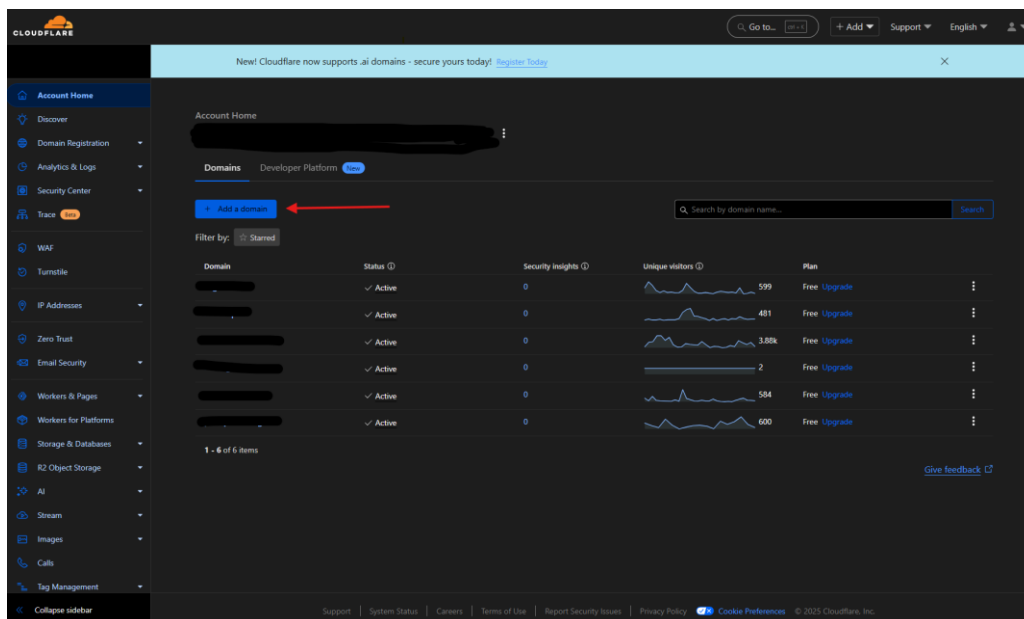
تصویر 2-4 ثبت نام در سایت کلودفلر

بعد از ثبت نام به پروفایل خود میریم



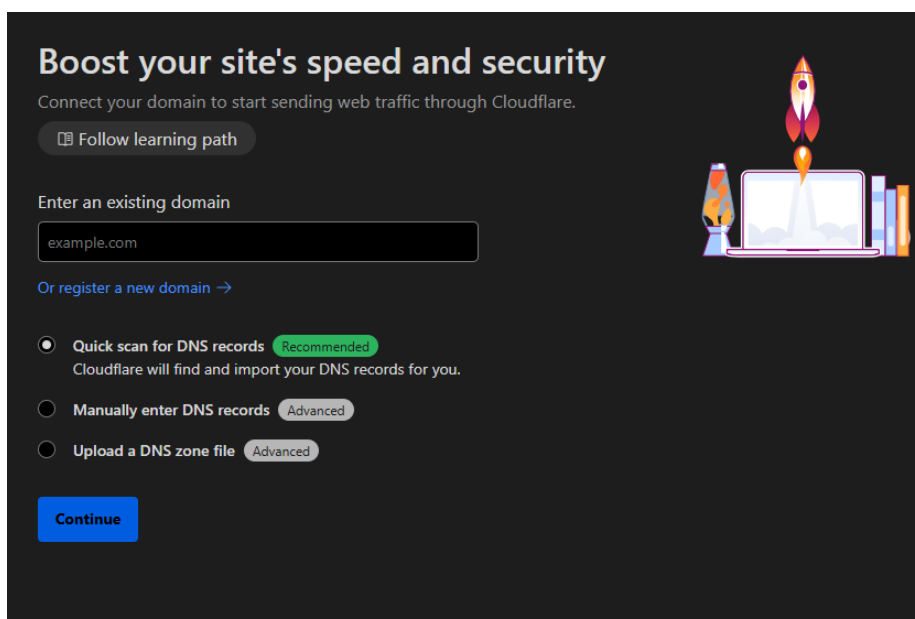
تصویر 2-5 وارد شدن به حساب کلودفلر

در پروفایل خود روی گزینه اضافه کردن دامنه کلیک میکنیم (ایمیل این اکانت و بقیه دامنه های ثبت شده در آن از تصویر حذف شده اند)



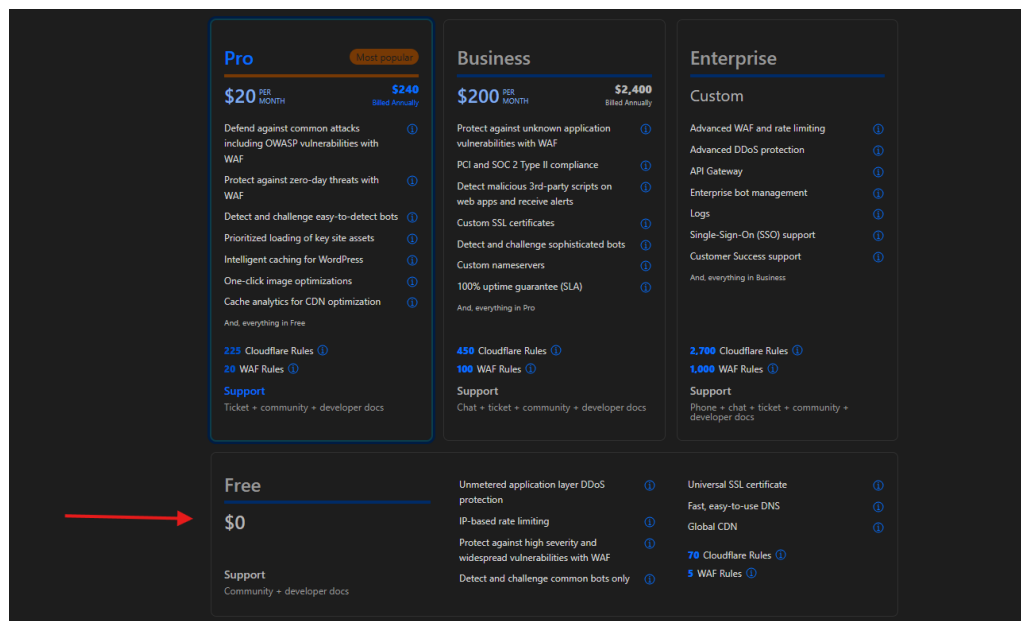
تصویر 2-6 اضافه کردن دامنه

در پنجره ظاهر شده دامنه خود را وارد میکنیم و روی گزینه ادامه کلیک میکنیم



تصویر 2-7 اضافه کردن دامنه

از بسته مجانی سایت استفاده میکنیم



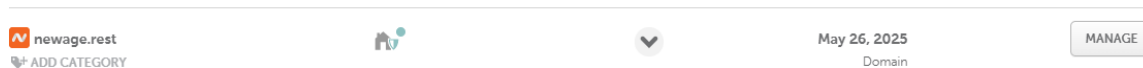
تصویر 2-8 انتخاب پلن کلودفلر

در نهایت باید نیم سرور های دامنه خود را با نیم سرور های داده شده توسط کلودفلر در سایت نیم چپ عوض کنیم نیم سرور های ظاهر شده در صفحه بعدی را کپی و به سایت نیم چپ برمیگردیم



تصویر 2-9 نیم سرورهای کلودفلر

در سایت نیم چپ وارد پروفایل شده و گزینه manage را میزنیم



تصویر 2-10 تنظیمات دامنه در نیم چپ

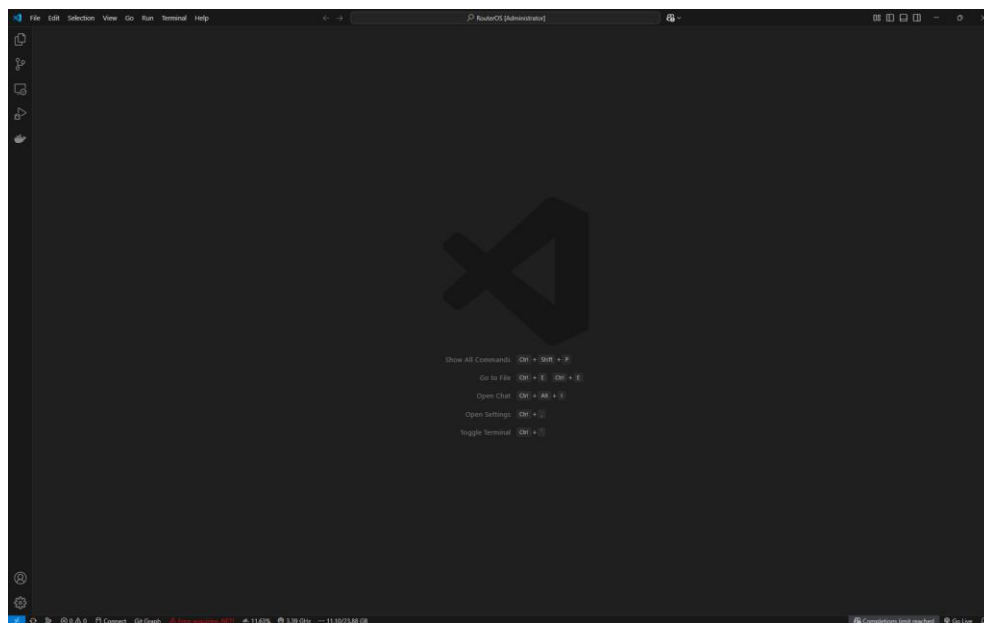
در قسمت نیم سرور آدرس ها را با آدرس هایی که از کلودفلر گرفتیم جاگذاری کرده و صبر میکنیم تا دامنه روی سایت کلودفلر فعال شود
تا اینجا فرآیند تامین دامنه به پایان رسید

4-2- هاست (host)

هاست به کامپیوتری اطلاق می شود که فایل های سایت روی آن قرار گرفته و فایل ها را به بینندگان ارائه میدهد

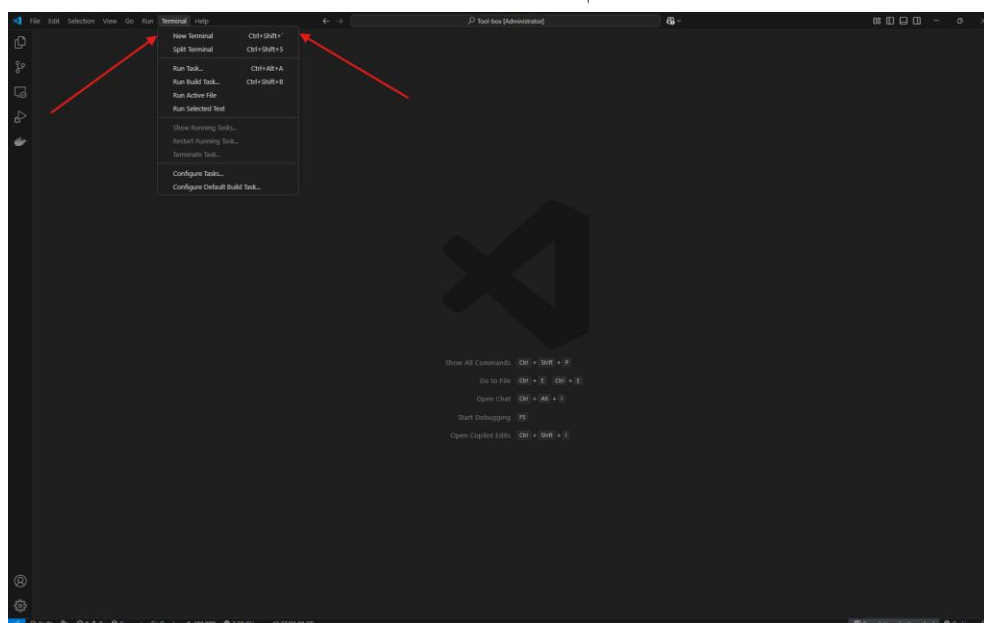
برای طراحی و نشر یک سایت راحت ترین اما لزوماً نه مقرون به صرفه ترین راه استفاده از سایت های ارائه میزبانی وردپرس هستند اصولاً مشکلی که در رابطه با این سایت ها وجود دارد اول گرانی این سایت ها هست چراکه برای کارهای ساده ای که توسط خود طراح سایت به سادگی قابل انجام است هزینه گرفته می شود مانند گرفتن گواهی اس اس ال یا فایروال و غیره است که توسط طراح به سادگی و بدون هزینه توسط راه کارهای اوپن سورس قابل پیاده سازی بدون هزینه است و دوم ضعف سخت افزاری در میزبانی سایت است چرا که در خیلی از این سایت های ارائه دهنده سایت پردازنده به شکل اشتراکی بین چند سایت استفاده می شود که به کاهش کارایی سایت منجر میشود و گزینه پردازنده اختصاصی در این سایت ها هزینه زیادی در بر دارد و سوم این است که برای افرادی که سایت خود را با ابزاری غیر از وردپرس طراحی می کنند این سایت ها گزینه مناسبی برای میزبانی سایت نمی باشد

شرکت های کلود (Cloud) معروف مانند AWS و Azure و Google cloud platform و Oracle cloud گزینه های مجانی برای هاست دارند شرکت اوراکل برای هر اکانت یک هاست با 4 هسته و 24 گیگ رم را به شکل مجانی ارائه میدهد که در این پروژه ما از این سرویس استفاده میکنیم هرچند ثبت نام در این سایت ها سخت است اما ما میتوانیم هاست های مناسب را از شرکت های داخلی مانند آروان کلود خریداری کنیم پس از خرید ما یک آدرس آی پی و یک پسورد دریافت میکنیم اگر در زمان خرید یک جفت کلید SSH مشخص کرده باشیم میتوانیم با کلید خصوص و آی پی به هاست خود متصل شویم
پس از نصب ویژوال استودیو کد با نصب افزونه Remote development میتوانیم به هاست خود متصل شویم
ابتدا ویژوال استودیو کد را باز میکنیم



تصویر 11-2 صفحه اصلی ویژوال استودیو کد

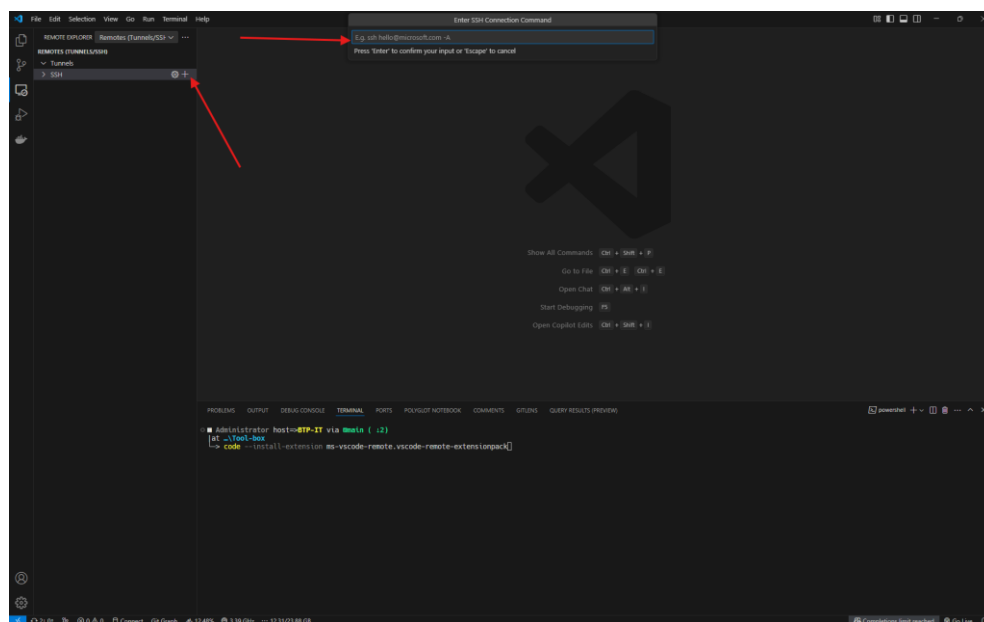
از قسمت ترمینال یک ترمینال جدید باز میکنیم



تصویر 12-2 باز کردن ترمینال جدید

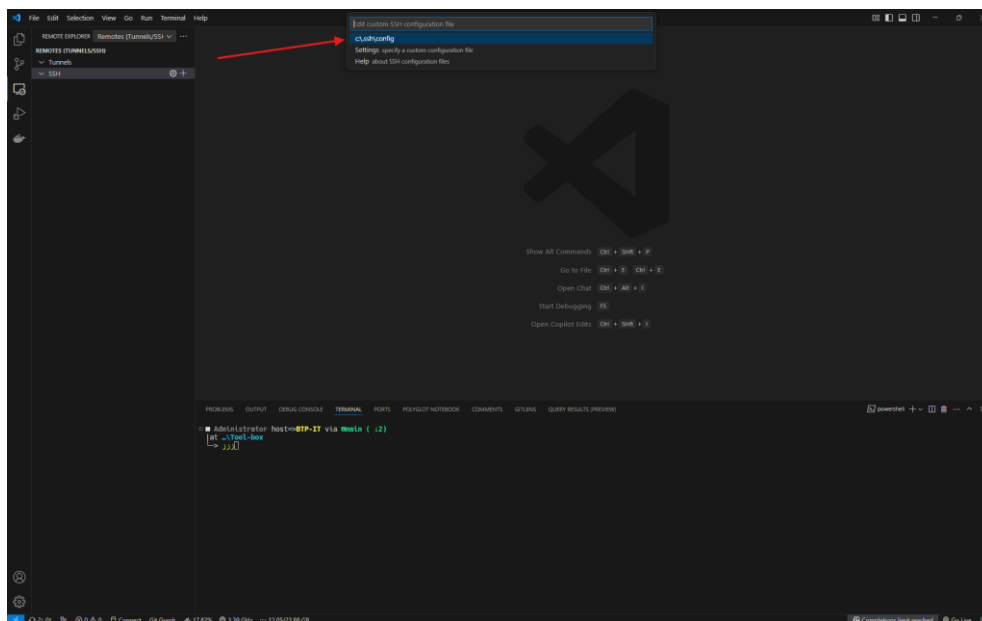
در ترمینال باز شده فرمان زیر را وارد میکنیم تا افزونه نصب شود
code --install-extension ms-vscode-remote.vscode-remote-extensionpack

سپس با کلیک روی علامت اضافه کردن مشخصات هاست خود را به شکل زیر اضافه میکنیم
user@ipaddress



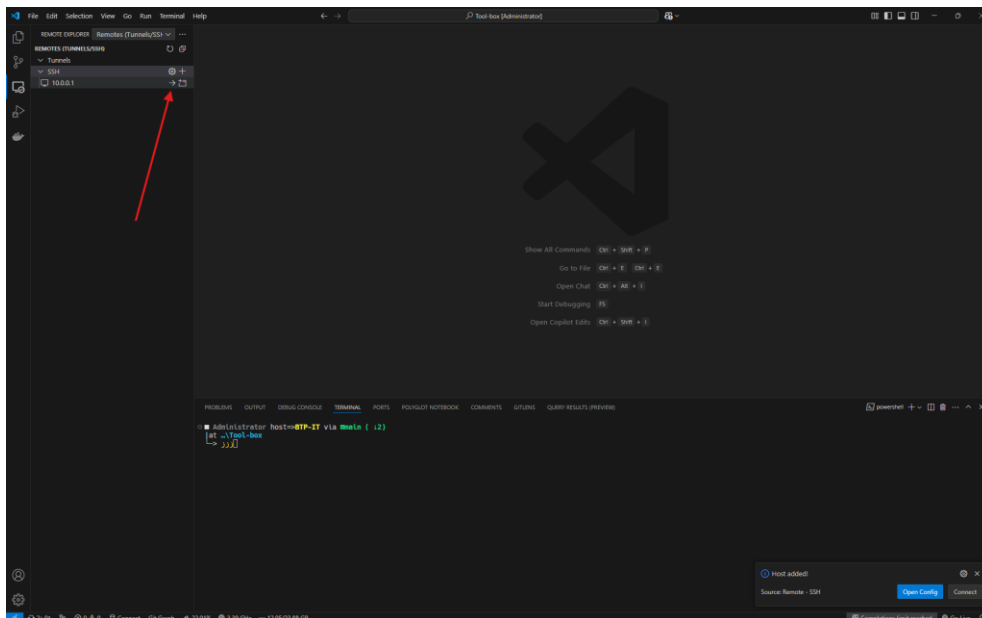
تصویر 15-2 اضافه کردن هاست

سپس روی محل ذخیره سازی پیش فرض کلیک میکنیم تا مشخصات کانکشن ما ذخیره شود



تصویر 16-2 اضافه کردن هاست

سپس می‌توانیم با کلیک روی گزینه اتصال به هاست خود متصل شویم



تصویر 17-2 اتصال به هاست

سپس پسورد داده شده توسط شرکتی که از آن هاست را خریده ایم وارد میکنیم

حالا ما توسط SSH به هاست خود متصل شده ایم و هاست ما آماده به استفاده است

فصل سوم

پیاده سازی سیستم

1-3- مقدمه

با توجه به طراحی سیستم و شناسایی نیازمندی ها در فصل 2 و شرح تامین آن ها در آن فصل. در این فصل سایت را در اینترنت منتشر میکنیم و به تامین امنیت و حفظ کیفیت سرویس دهی میپردازیم.

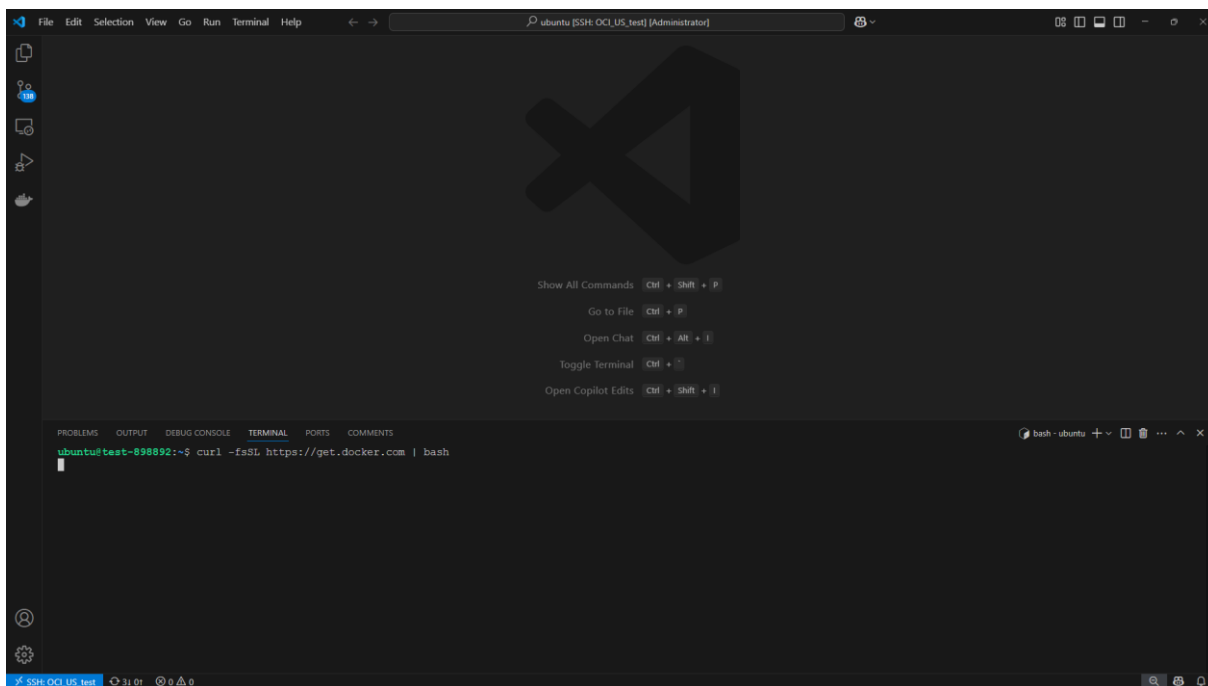
2-3- نصب داکر (Docker)

داکر یک نرم افزار مجازی سازی است به این معنی که داکر روی سخت افزار سیستم ما نصب شد و برای هر نرم افزار دیگری که با استفاده از داکر پیاده سازی شود یک محیط مجازی جدا از سیستم عامل سخت افزار ایجاد میکند این ویژگی به ما کمک میکند که سرویس های مورد نیاز خود را بدون نگرانی از اختلال با سرویس های دیگر به آسانی پیاده سازی کنیم به علاوه سرویس هایی که به این شکل پیاده سازی می شوند همان طور که در آینده خواهیم دید به آسانی قابلیت پیاده سازی در سیستم های دیگر و تغییر زیر ساخت سرویس را بدون از دسترس خارج شدن آن سرویس دارد

در فصل قبل متوجه شدیم که چگونه به وسیله پروتکل SSH به هاست خود متصل شویم اکنون داکر را روی هاست خود نصب می کنیم

یک ترمینال جدید در نرم افزار باز کرده و دستور زیر را در آن اجرا میکنیم تا داکر روی آن نصب شود

```
curl -fsSL https://get.docker.com | bash
```

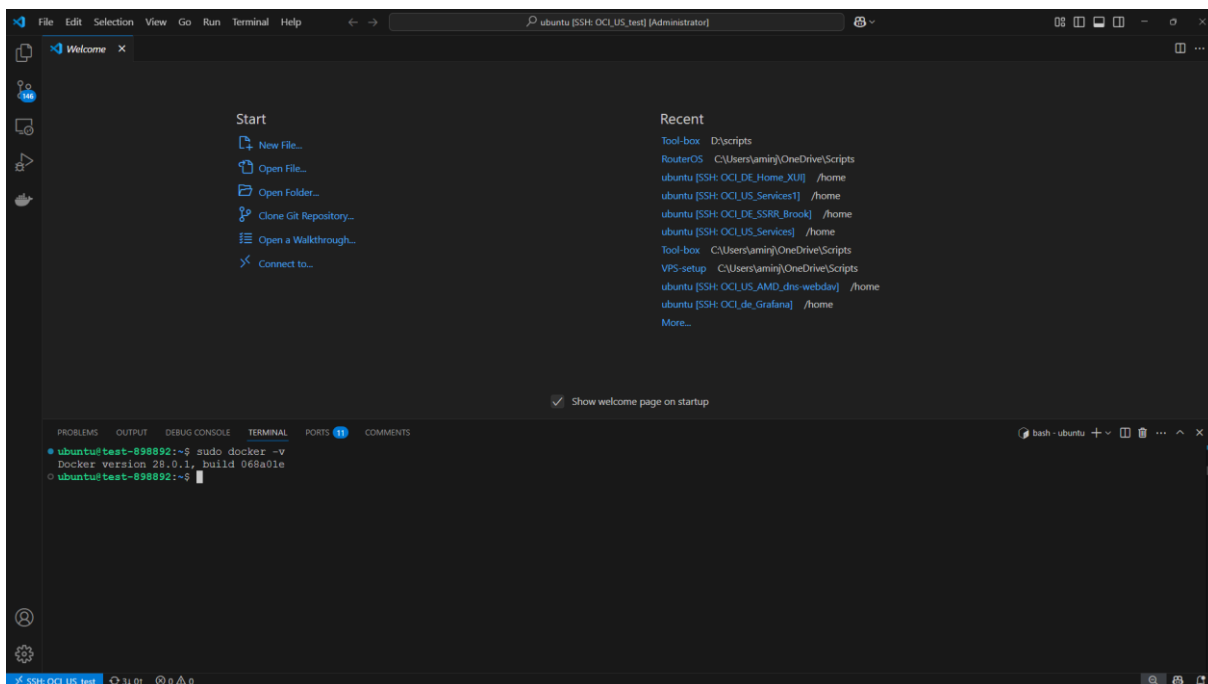


تصویر 1-3 نصب داکر

پس از آن با اجرای فرمان

```
sudo docker -v
```

تست میکنیم که داکر بدون مشکل نصب شده باشد اگر این فرمان بدون ارور اجرا و ورژن داکر نمایش داده شود یعنی داکر بدون مشکل روی سیستم عامل ما نصب شده است



تصویر 2-3 تست داکر

3-3- فایل داکر کامپوس (Docker Compose)

فایل داکر یک فایل مشخصات است که مشخص می کند سرویس ما با چه مشخصاتی اجرا شود در فایل زیر ما وردپرس را با مشخصات زیر اجرا می کنیم که در ادامه به توضیح هر کدام از کدهای زیر می پردازیم

```
services:
  wordpress:
    depends_on:
      - wordpress_db
    container_name: WordPress
    hostname: wordpress
    image: wordpress
    restart: always
    ports:
      - 8000:80
    environment:
      WORDPRESS_DB_HOST: wordpress_db
      WORDPRESS_DB_USER: ${WordPress_db_user}
      WORDPRESS_DB_PASSWORD: ${WordPress_Pass}
      WORDPRESS_DB_NAME: WordPress_db
    volumes:
      - ./WordPress:/var/www/html
    networks:
      - my_network
    healthcheck:
      test: ["CMD", "curl", "--fail", "http://localhost:80"]
```

```

    interval: 1m30s
    timeout: 30s
    retries: 5
    start_period: 30s

wordpress_db:
  container_name: wordpress_db
  image: mysql:8.0
  restart: always
  environment:
    MYSQL_DATABASE: WordPress_db
    MYSQL_USER: ${WordPress_db_user}
    MYSQL_PASSWORD: ${WordPress_Pass}
    MYSQL_RANDOM_ROOT_PASSWORD: "1"
  healthcheck:
    test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
    interval: 1m30s
    timeout: 30s
    retries: 5
    start_period: 30s
  volumes:
    - ./WordPress_database:/var/lib/mysql
  networks:
    - my_network

wordpress_redis:
  image: redis
  container_name: wordpress_redis
  hostname: wordpress_redis
  restart: always
  expose:
    - 6379
  volumes:
    - ./WordPress_redis/redis.conf:/usr/local/etc/redis/redis.conf
  command: redis-server /usr/local/etc/redis/redis.conf
  networks:
    - my_network
  healthcheck:
    test: ["CMD", "redis-cli", "ping"]
    interval: 30s
    timeout: 10s
    retries: 5

networks:
  my_network:
    name: my_network
    driver: bridge

```

3-3-1 سرویس WordPress

این سرویس همان سیستم مدیریت محتوای وردپرس است که روی یک کانتینر اجرا می‌شود.

جزئیات:

depends_on:

- wordpress_db

مشخص می‌کند که این سرویس بعد از راه‌اندازی دیتابیس (wordpress_db) اجرا شود.

container_name: WordPress

hostname: wordpress

نام کانتینر را WordPress و نام میزبان (hostname) را wordpress تنظیم می‌کند.

image: wordpress

از ایمج رسمی wordpress در داکر هاب استفاده می‌کند.

restart: always

اگر کانتینر کرش کند، داکر آن را دوباره اجرا خواهد کرد.

ports:

- 8000:80

پورت ۸۰ هاست را به کانتینر متصل می‌کند. مقدار پورت سمت هاست مشخص نشده که یعنی یک پورت

تصادفی اختصاص داده می‌شود.

environment:

WORDPRESS_DB_HOST: wordpress_db

WORDPRESS_DB_USER: \${WordPress_db_user}

WORDPRESS_DB_PASSWORD: \${WordPress_Pass}

WORDPRESS_DB_NAME: WordPress_db

متغیرهای محیطی برای ارتباط وردپرس با دیتابیس. مقادیر WordPress_db_user و WordPress_Pass

از env گرفته می‌شوند.

volumes:

- ./WordPress:/var/www/html

پوشه ./WordPress در سیستم میزبان را به /var/www/html در داخل کانتینر متصل می‌کند. این باعث

می‌شود که فایل‌های وردپرس در میزبان ذخیره شوند و با حذف کانتینر از بین نروند.

networks:

- my_network

این کانتینر را به شبکه‌ای به نام my_network متصل می‌کند.

healthcheck:

test: ["CMD", "curl", "--fail", "http://localhost:80"]

interval: 1m30s

timeout: 30s

retries: 5

start_period: 30s

بررسی می‌کند که آیا وردپرس روی پورت ۸۰ پاسخ می‌دهد یا نه.

1-4-1- سرویس دیتابیس (wordpress_db)

container_name: wordpress_db

image: mysql:8.0

restart: always

این سرویس از ایمج رسمی mysql:8.0 برای اجرای یک دیتابیس MySQL استفاده می‌کند.

در صورت کرش کردن، مجدداً راه‌اندازی خواهد شد.

environment:

MYSQL_DATABASE: WordPress_db

MYSQL_USER: \${WordPress_db_user}

MYSQL_PASSWORD: \${WordPress_Pass}

MYSQL_RANDOM_ROOT_PASSWORD: "1"

مقادیر محیطی برای تنظیم دیتابیس:

- MYSQL_DATABASE: نام دیتابیس
- MYSQL_USER: نام کاربری که از env خوانده می‌شود
- MYSQL_PASSWORD: پسورد که از env خوانده می‌شود
- MYSQL_RANDOM_ROOT_PASSWORD: یک رمز تصادفی برای کاربر root تنظیم می‌شود (چون نیازی به آن نداریم).

volumes:

- ./WordPress_database:/var/lib/mysql

ذخیره‌سازی داده‌های دیتابیس در مسیر ./WordPress_database روی هاست.

networks:

- my_network

این سرویس را به شبکه my_network متصل می‌کند.

healthcheck:

test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]

interval: 1m30s

timeout: 30s

retries: 5

start_period: 30s

بررسی می‌کند که دیتابیس فعال است و در شرایط مناسب فعالیت می‌کند یا نه.

2-1-4- سرویس کشینگ ردیس

ما از ردیس برای افزایش سرعت سایت استفاده میکنیم

```
image: redis
container_name: wordpress_redis
hostname: wordpress_redis
restart: always
```

این سرویس یک سرور **Redis** اجرا می کند که برای کشینگ وردپرس استفاده می شود.

```
expose:
  - 6379
```

پورت 6379 را در شبکه داخلی داکر در دسترس قرار می دهد.

```
volumes:
  - ./WordPress_redis/redis.conf:/usr/local/etc/redis/redis.conf
```

فایل کانفیگ `redis.conf` از هاست به داخل کانتینر منتقل می شود.

```
command: redis-server /usr/local/etc/redis/redis.conf
```

Redis را با استفاده از فایل کانفیگ بالا اجرا می کند.

```
networks:
  - my_network
```

این سرویس را به `my_network` متصل می کند.

```
healthcheck:
  test: ["CMD", "redis-cli", "ping"]
  interval: 30s
  timeout: 10s
  retries: 5
```

تست می کند که آیا Redis به درستی کار می کند یا نه.

4-1-3- شبکه ها (Networks)

```
networks:
  my_network:
    name: my_network
    driver: bridge
```

یک شبکه به نام my_network با درایور bridge تعریف شده است. همه سرویس‌ها در همین شبکه ارتباط دارند.

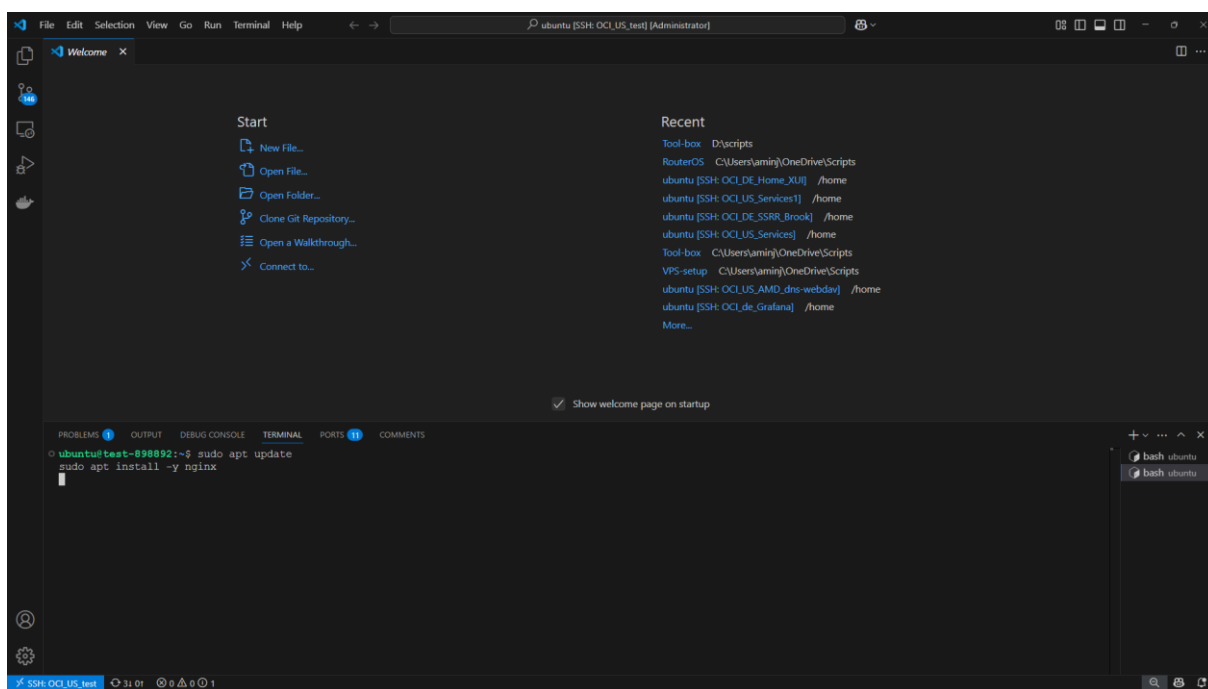
4-2- نصب nginx

برای نشر سایت خود با گواهی SSL ما به یک نرم افزار Reverse proxy نیاز داریم تا گواهی SSL را بتوانیم به مرورگر بینندگان خود عرضه کنیم تا سایت ما به عنوان یک سایت امن در فضای اینترنت قرار بگیرد

برای نصب nginx در یک ترمینال در هاست خود فرمان زیر را اجرا میکنیم

```
sudo apt update
```

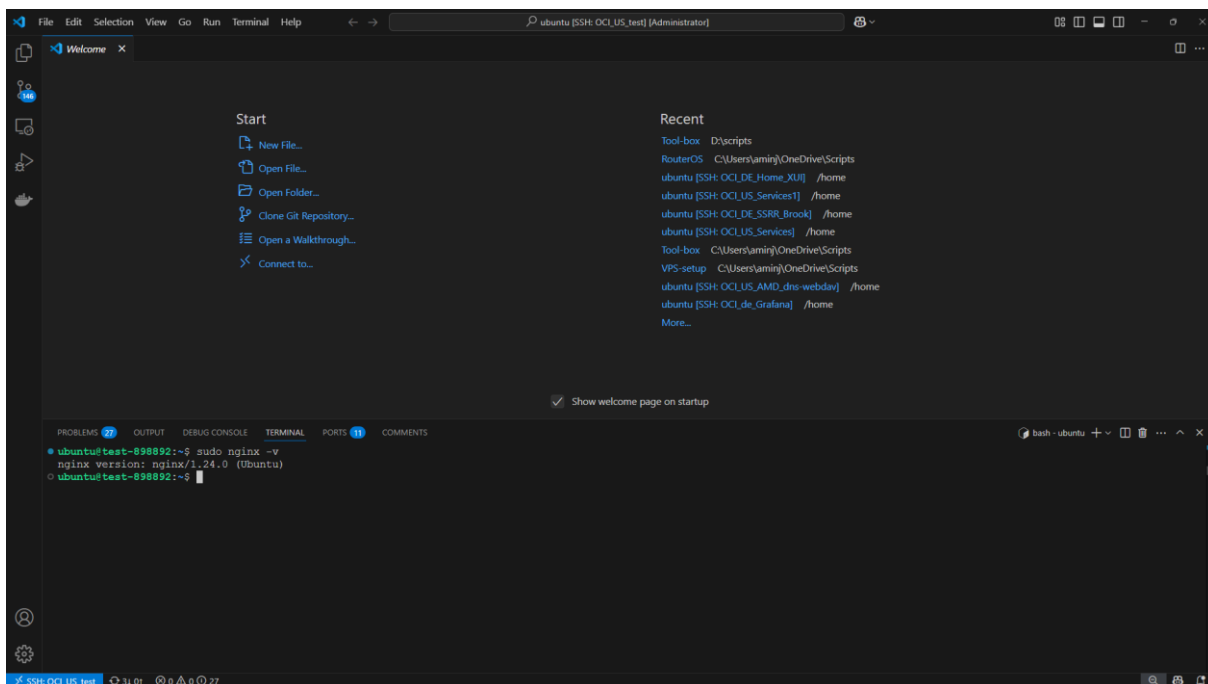
```
sudo apt install -y nginx
```



تصویر 3-3 نصب nginx

پس از اجرای فرمان های بالا ما تست میکنیم تا ببینیم nginx به درستی نصب شده است یا نه

```
nginx -v
```



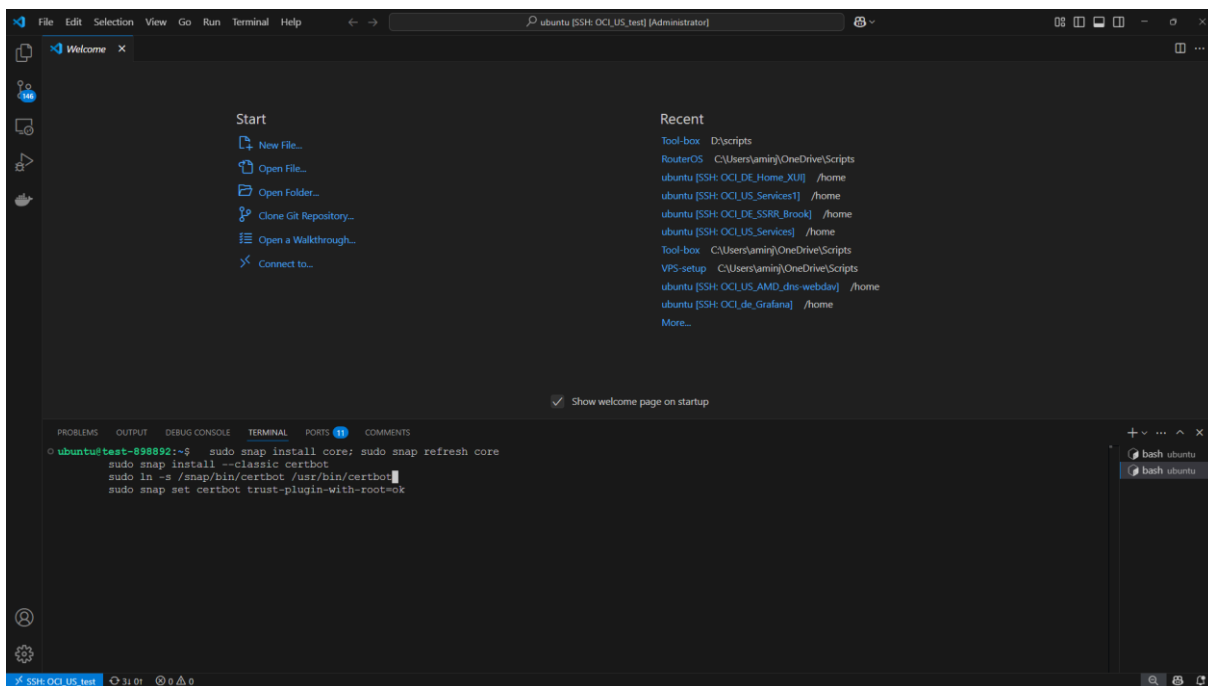
تصویر 3-4 تست nginx

حالا با نصب nginx در ادامه با استفاده از certbot گواهی SSL دریافت میکنیم

4-3- نصب certbot

این نرم افزار به ما کمک میکند تا گواهی SSL را دریافت و آن را تمدید کنیم
با اجرای فرمان زیر certbot را نصب میکنیم

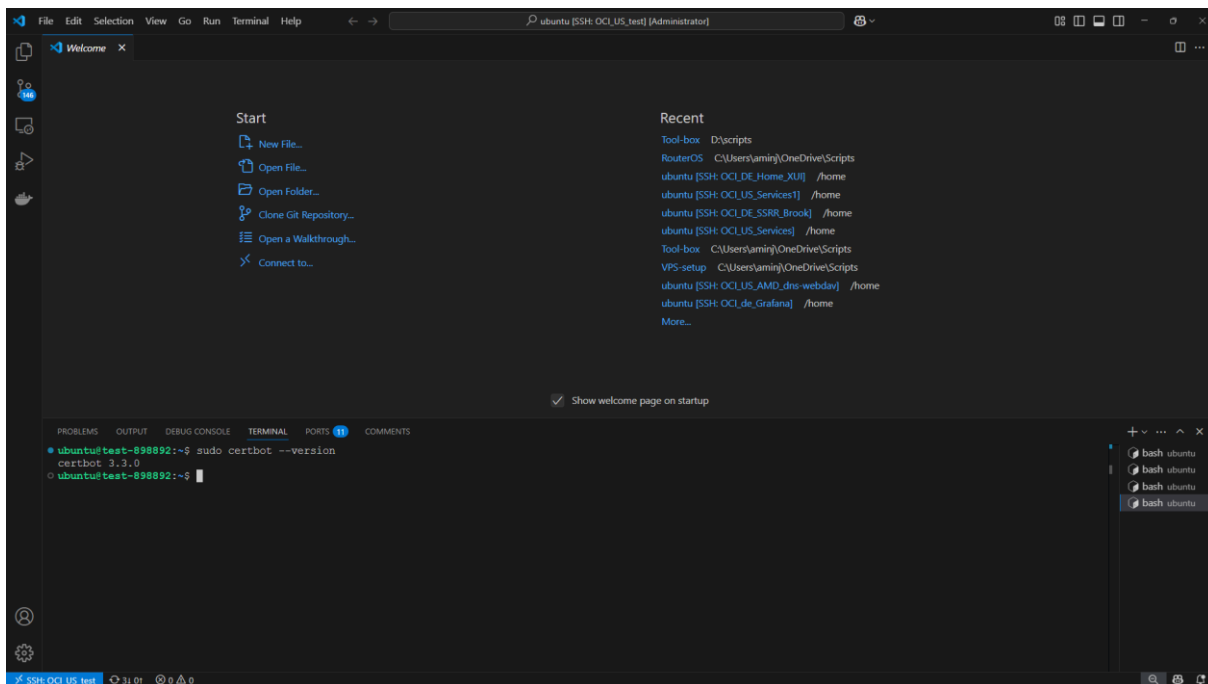
```
sudo snap install core; sudo snap refresh core
sudo snap install --classic certbot
sudo ln -s /snap/bin/certbot /usr/bin/certbot || true
sudo snap set certbot trust-plugin-with-root=ok
```



تصویر 3-5 نصب nginx

پس از اجرای فرمان های فوق فرمان زیر را اجرا میکنیم تا مطمئن شویم certbot نصب شده و به درستی کار میکند

`sudo certbot --version`

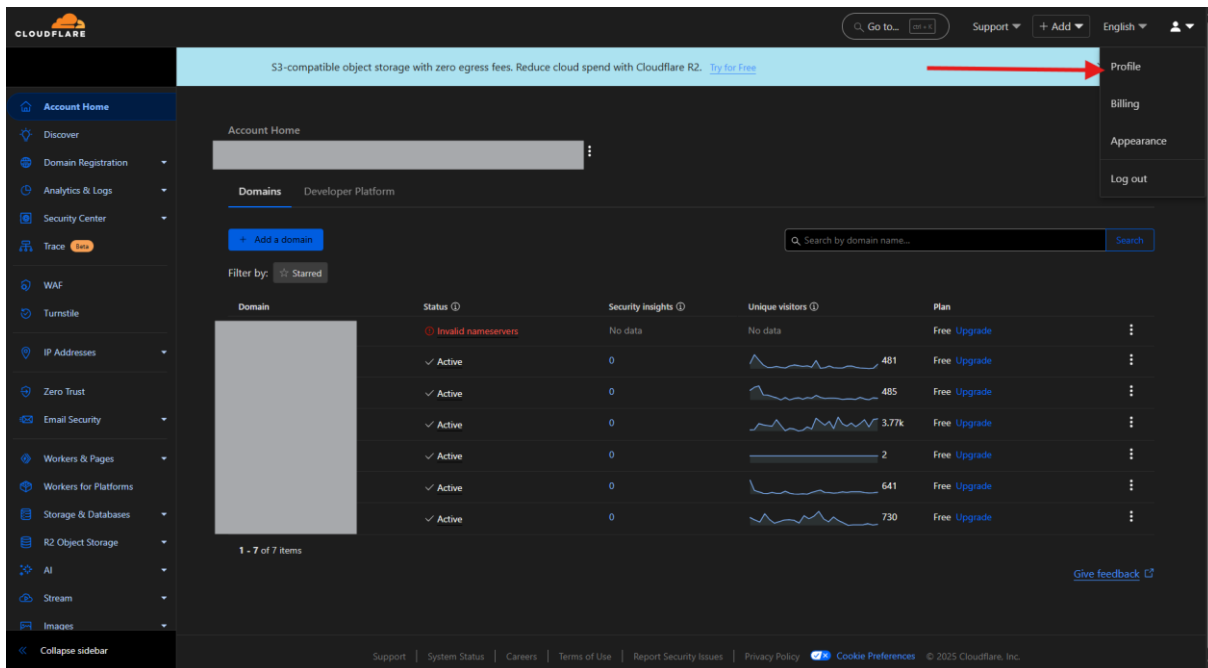


تصویر 3-6 تست certbot

4-4- گرفتن توکن کلودفلر

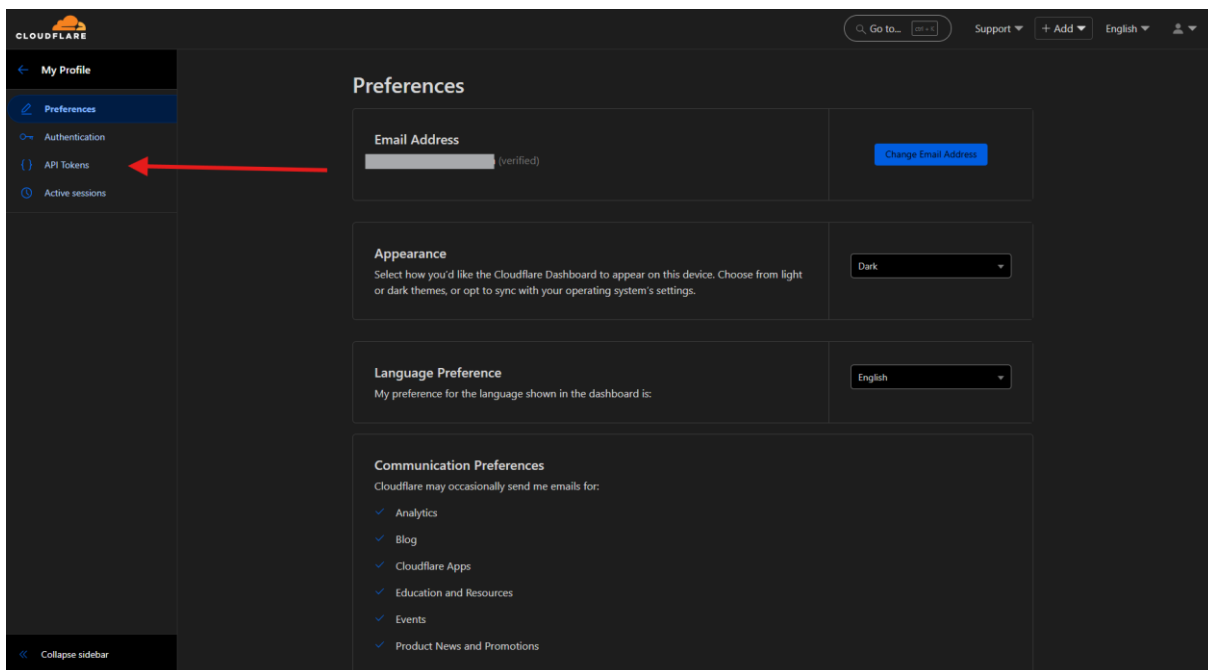
حالا که certbot نصب شده است میتوانیم از آن استفاده کنیم تا برای دامنه خود گواهی SSL دریافت کنیم در ابتدا یک کلید API از سایت کلودفلر برای دامنه خود دریافت کرده و از آن و certbot گواهی SSL دریافت کنیم

وارد حساب کلودفلر خود میشویم و به قسمت اکانت و بخش پروفایل میشویم



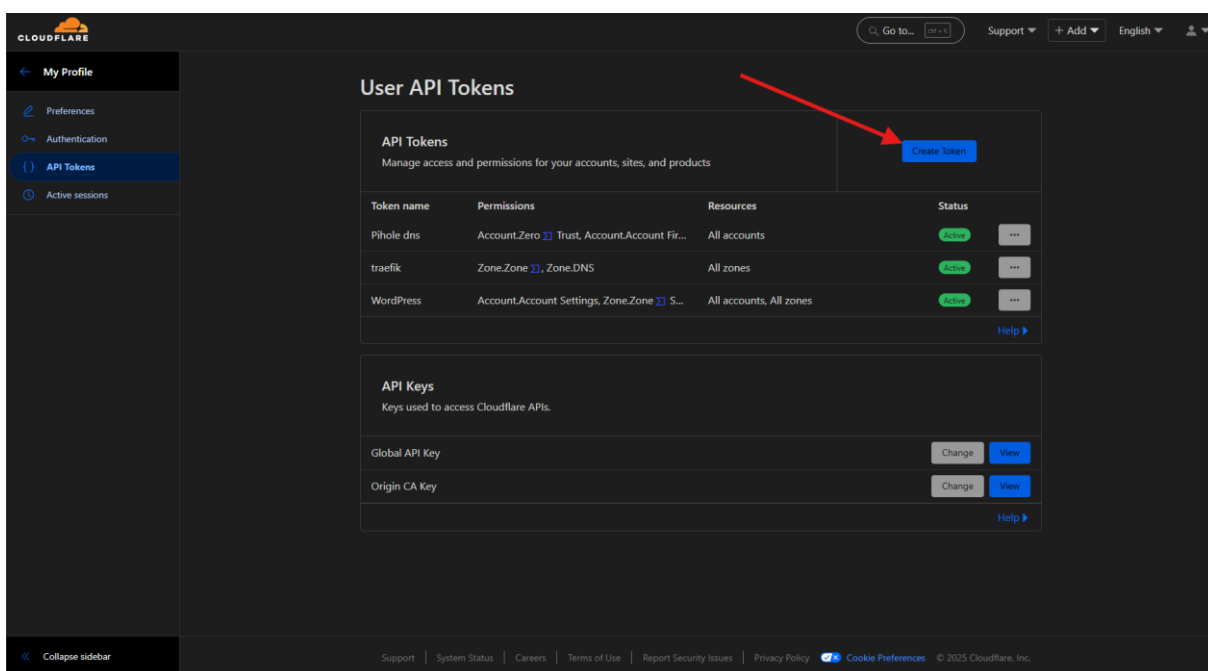
تصویر 3-7 قسمت پروفایل اکانت کلودفلر

به قسمت API token میرویم



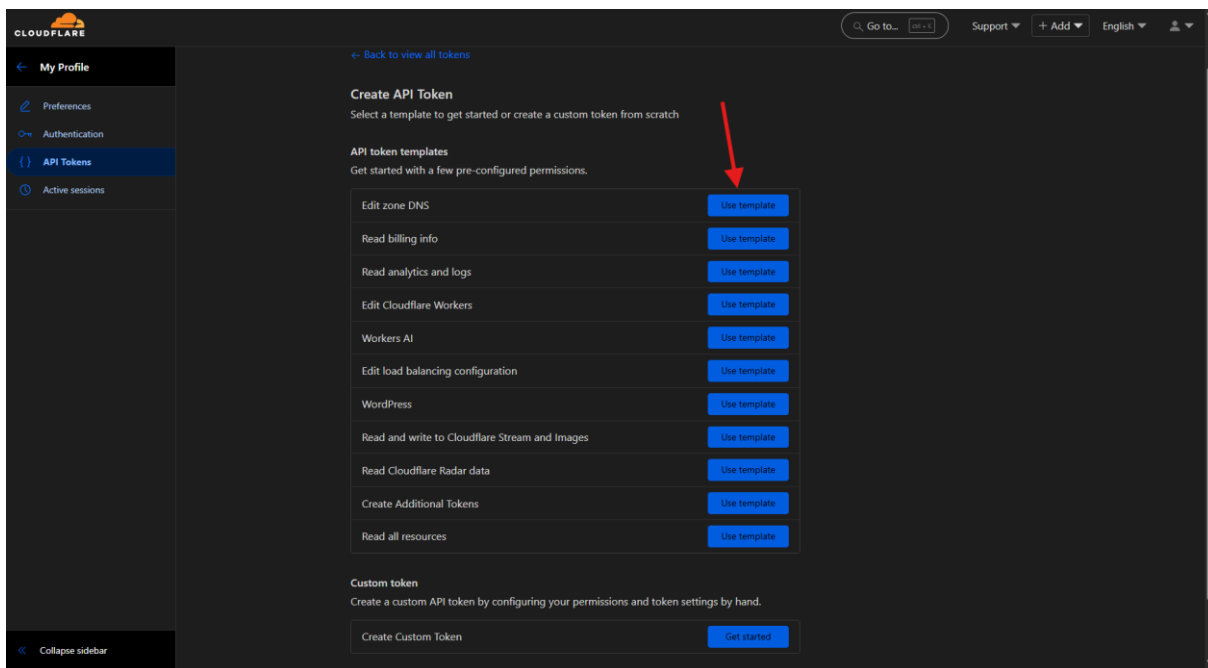
تصویر 3-8 API token

روی دکمه ساخت توکن کلیک میکنیم

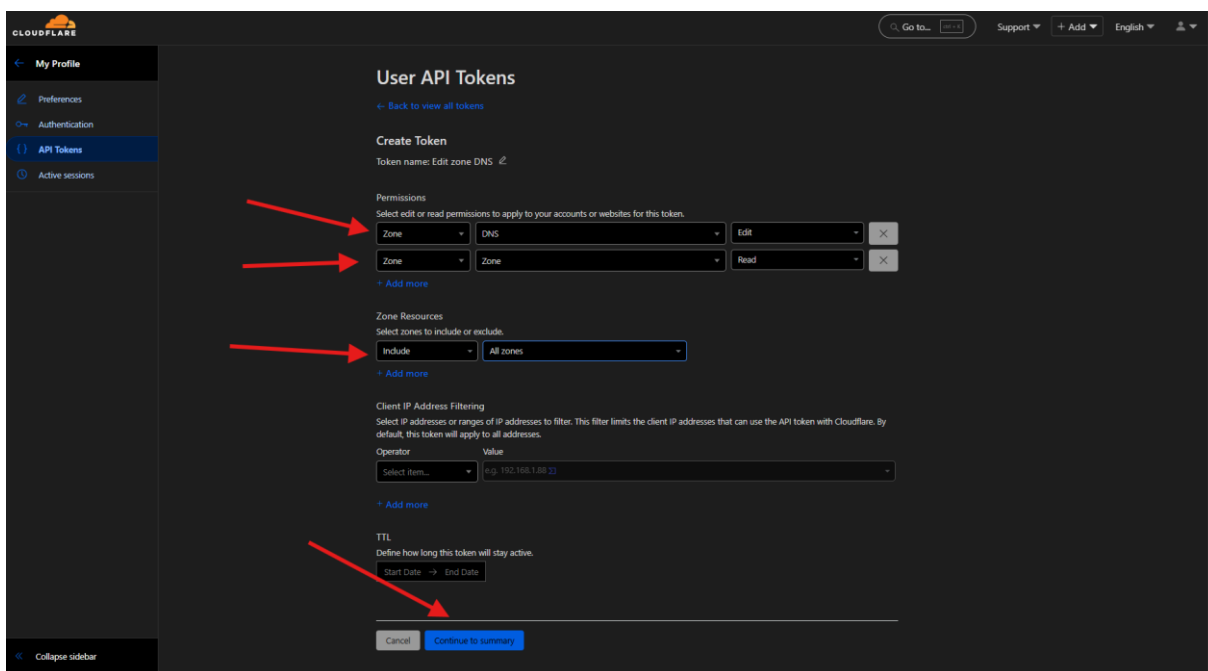


تصویر 3-9 ساخت توکن

توکن را مطابق شکل زیر با دسترسی های ذکر شده میسازیم



تصویر 10-3 توکن کلودفلر



تصویر 11-3 ساخت توکن

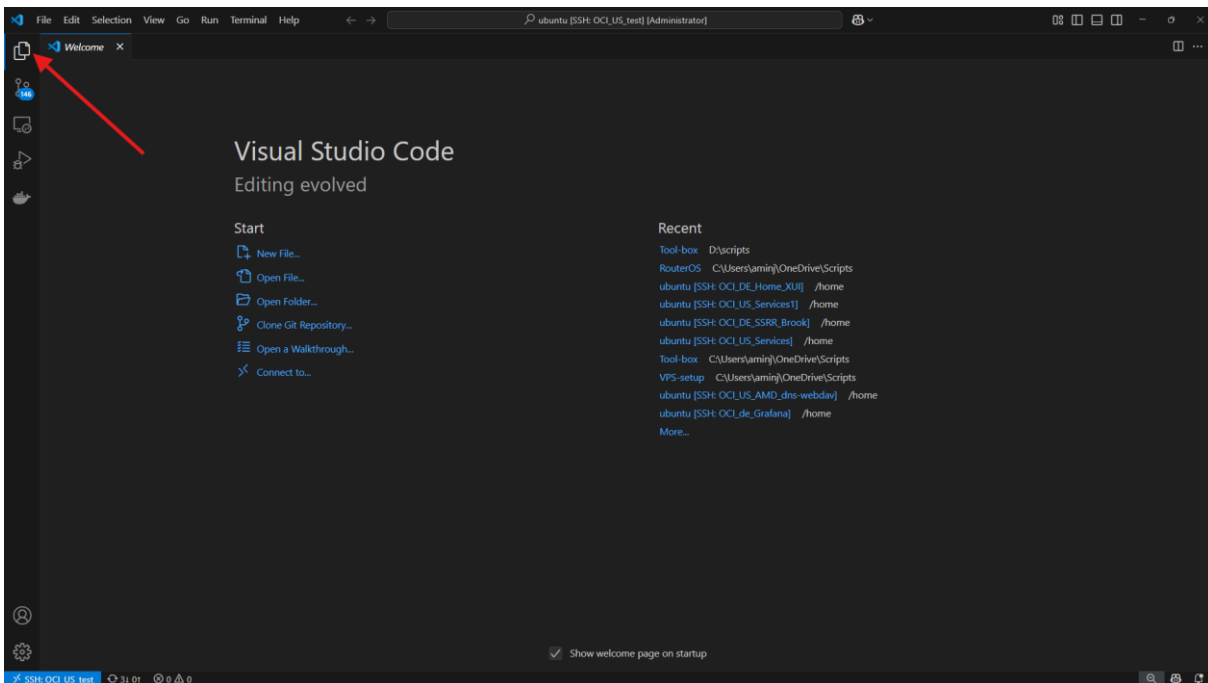
روی دکمه ساخت در صفحه بعد کلیک کرده و کد ظاهر شده را کپی میکنیم این کد api لازم برای گرفتن گواهی SSL است

حال به صفحه اصلی کاربری برگشته روی دامنه خود کلیک کرده در نوار سمت چپ روی DNS کلیک کرده روی دکمه add record کلیک کرده و دامنه سایت خود را وارد و آدرس IP هاست خود را وارد میکنیم تا دامنه به هاست ما اشاره کند.

Wordpress اجرای سرویس

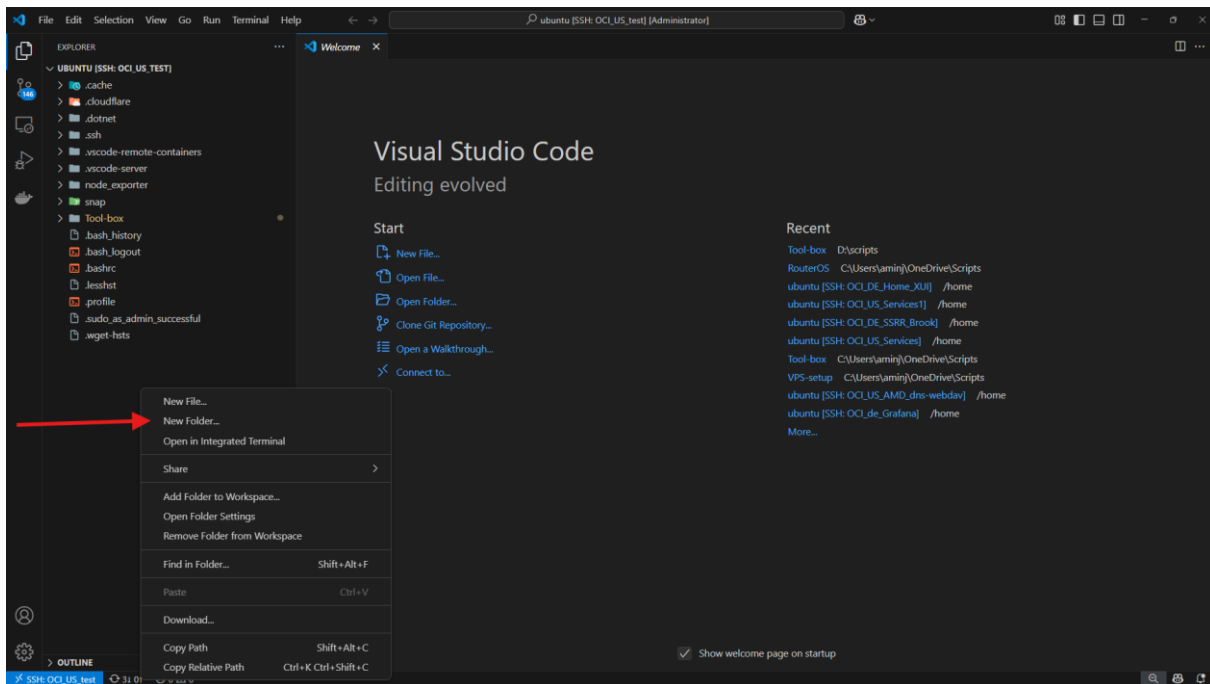
حالا که تمام پیش نیازهای ما برآورده شده اند وقت آن است که سرویس وردپرس را به کمک داکر اجرا و سپس در مراحل بعدی سایت را با گواهی SSL در فضای اینترنت منتشر و به بهینه کردن آن پردازیم یک فولدر به اسم wordpress ایجاد کرده و یک فایل به اسم docker-compose.yml ایجاد و محتویات فایل را که در مرحله 3-3 در آن کپی میکنید

روی دکمه explorer کلیک کنید تا فایل ها و فولدرها نمایش داده شود



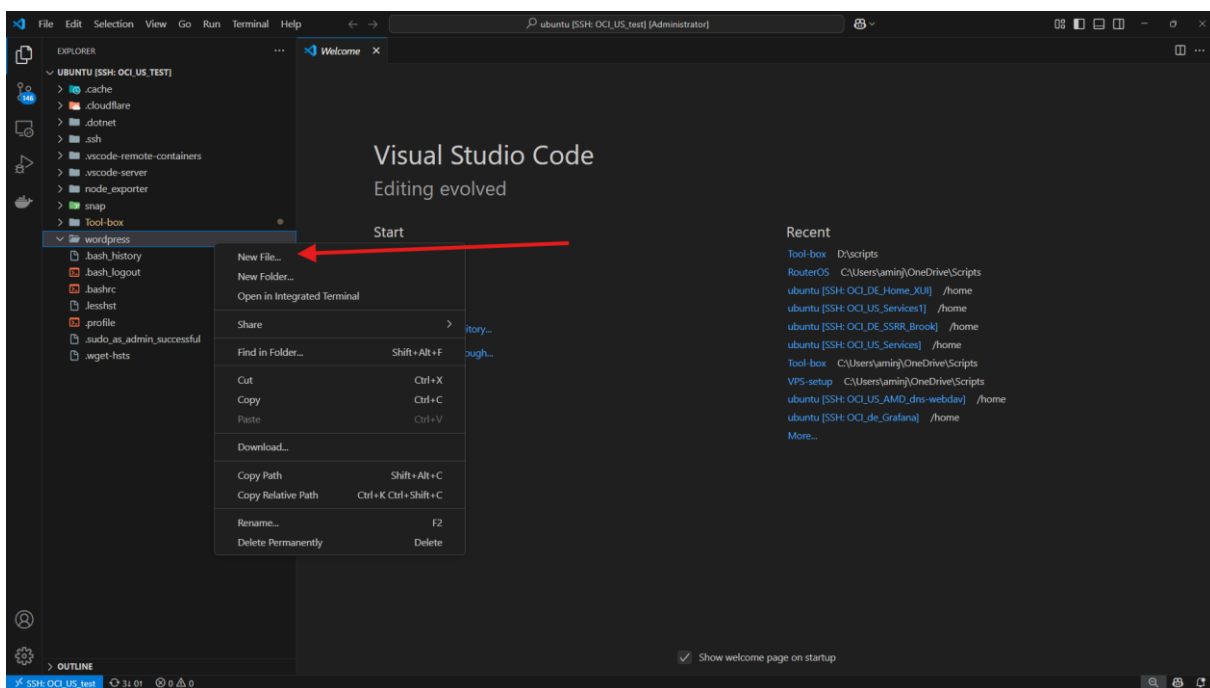
تصویر 12-3 vscode explorer

کلیک راست کرده و یک فولدر میسازیم



تصویر 13-3 ساخت فولدر

روی فولدر wordpress کلیک راست کرده و یک فایل جدید میسازیم



تصویر 143- ساخت فایل

در فایل جدید پس از کپی کردن محتویات به جای `{WordPress_db_user}$` یک نام کاربری مناسب و به جای `{WordPress_Pass}$` یک پسورد مناسب بگذارید پس از آن با اجرای فرمان های زیر سرویس وردپرس را اجرا میکنیم

```
cd wordpress
```

```
sudo docker compose up -d
```

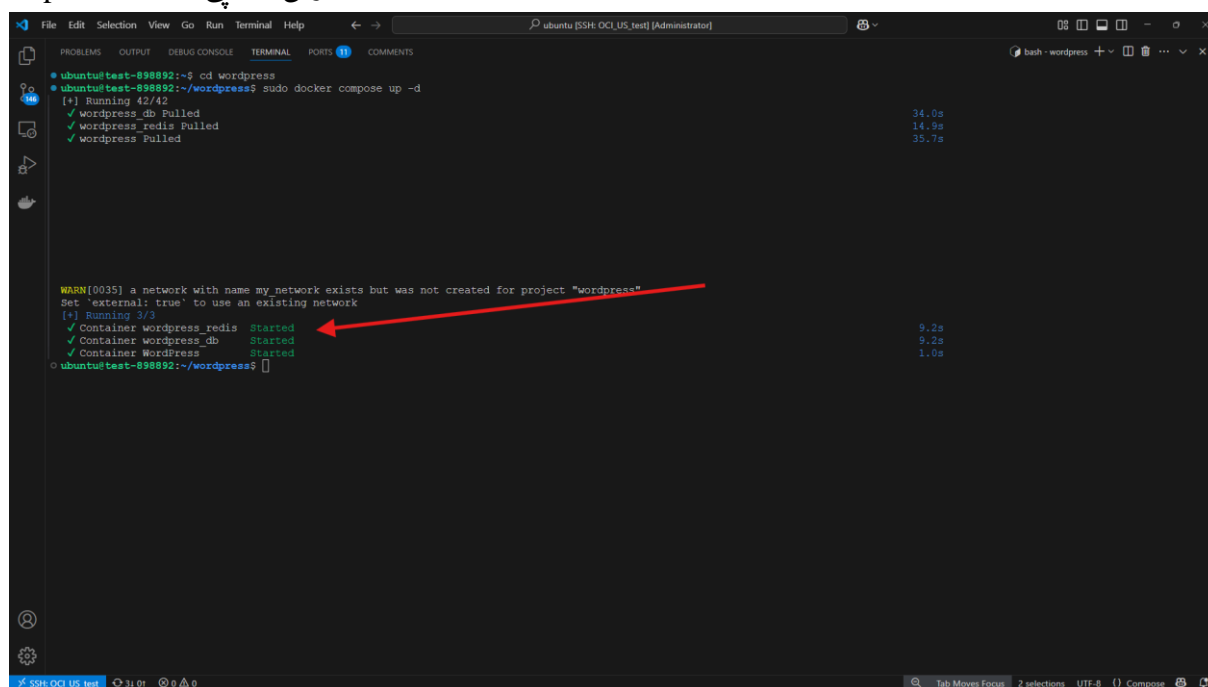
حالا باید پورت های 80 و 443 را هم در هاست خود باز بگذاریم تا سایت خود با پروتکل های HTTP و HTTPS در دسترس باشد برای این کار فرمان های زیر را در ترمینال اجرا میکنیم

```
sudo iptables -I INPUT -p tcp -j ACCEPT --dport 80
```

```
sudo iptables -I INPUT -p tcp -j ACCEPT --dport 443
```

حالا سرویس وردپرس ما با اتصال HTTP در آدرس زیر با استفاده از مرورگر در دسترس است

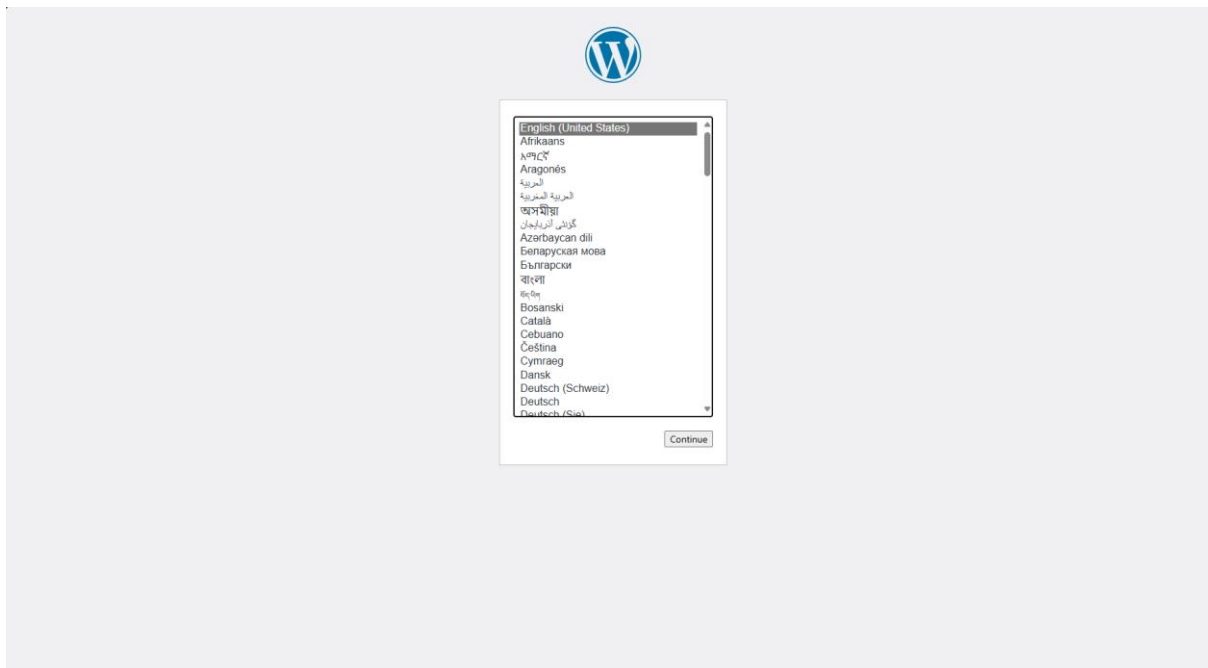
`http://<آدرس آی پی هاست>:8000`



```
ubuntu@test-898892:~$ cd wordpress
ubuntu@test-898892:~/wordpress$ sudo docker compose up -d
[*] Running 42/42
✓ wordpress_db Pulled                                34.0s
✓ wordpress_redis Pulled                             14.9s
✓ wordpress Pulled                                   35.7s

WARN[0035] a network with name my_network exists but was not created for project "wordpress"
Set 'external: true' to use an existing network
[*] Running 3/3
✓ Container wordpress_redis Started                  9.2s
✓ Container wordpress_db Started                     9.2s
✓ Container WordPress Started                       1.0s
ubuntu@test-898892:~/wordpress$
```

تصویر 15-3 اجرای سرویس وردپرس



تصویر 3-16 صفحه شروع وردپرس

حالا که میدانیم سرویس وردپرس به درستی شروع به کار کرده مرحله بعدی گرفتن گواهی SSL و انتشار آن با گواهی SSL به وسیله nginx است.

1-4-4- انتشار سایت با SSL به وسیله nginx و certbot

در درجه اول باید یک گواهی SSL با استفاده از certbot بگیریم برای این کار فرمان زیر را در ترمینال اجرا میکنیم که جای توکن و دامنه را با توکن و دامنه خود پر میکنیم

```
mkdir ~/.cloudflare
touch ~/.cloudflare/credentials.ini
echo "dns_cloudflare_api_token = $(CLOUDFLARE_API_TOKEN)" >
~/.cloudflare/credentials.ini
chmod 600 ~/.cloudflare/credentials.ini
sudo snap install certbot-dns-cloudflare
sudo certbot certonly --dns-cloudflare --agree-tos --no-eff-email --dns-cloudflare --dns-
cloudflare-credentials ~/.cloudflare/credentials.ini -d $(DOMAIN)
```

```
File Edit Selection View Go Run Terminal Help
ubuntu [SSH: OCL_US_test] [Administrator]
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
ubuntu@test-898892:~$ mkdir ~/.cloudflare
ubuntu@test-898892:~$ touch ~/.cloudflare/credentials.ini
ubuntu@test-898892:~$ echo "dns_cloudflare_api_token = $(CLOUDFLARE_API_TOKEN)" > ~/.cloudflare/credentials.ini
ubuntu@test-898892:~$ chmod 600 ~/.cloudflare/credentials.ini
ubuntu@test-898892:~$ sudo snap install certbot-dns-cloudflare
ubuntu@test-898892:~$ sudo certbot certonly --dns-cloudflare --agree-tos --no-eff-email --dns-cloudflare --dns-cloudflare-credentials ~/.cloudflare/credentials.ini -d $(DOMAIN)
```

تصویر 3-17 گرفتن گواهی SSL

```
Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/test.pixelprofit.org/fullchain.pem
Key is saved at: /etc/letsencrypt/live/test.pixelprofit.org/privkey.pem
This certificate expires on 2025-06-25.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
-----
ubuntu@test-898892:~$
```

تصویر 3-18 پیغام موفقیت گرفتن گواهی SSL

همانطور که در تصویر بالا مشخص است پیغام موفقیت دریافت گواهی SSL نمایش داده شد و مکان ذخیره سازی کلیدها در هاست نمایش داده شده

4-4-2 تنظیم NGINX

حالا که سرویس وردپرس اجرا شده و گواهی SSL هم داریم باید با استفاده از nginx سایت وردپرس خود را با HTTPS اجرا کنیم برای این کار ابتدا با فرمان زیر فایل تنظیمات nginx را باز میکنیم

code /etc/nginx/nginx.conf

در فایل باز شده تنظیمات پیش فرض را حذف و تنظیمات زیر را وارد میکنیم

```
events {
    worker_connections 1024;
}

http {
    # HTTP to HTTPS redirection
    server {
        listen 80;
        server_name domain.com;
        # Set webroot for certbot to issue certificates
        location /.well-known/acme-challenge/ {
            root /var/www/certbot;
        }
        #! For redirect to work cloudflare ssl must be set to full(strict)
        # Redirect http traffic to https traffic
        location / {
            return 301 https://$host$request_uri;
        }
    }

    # HTTPS server with reverse proxy
    server {
        listen 443 ssl;
        server_name domain.com;
        # Certificates issued by certbot
        ssl_certificate /etc/letsencrypt/live//domain.com/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live//domain.com/privkey.pem;

        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_prefer_server_ciphers on;
        ssl_ciphers "ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-
AES256-GCM-SHA384";

        location / {
            proxy_pass http://172.18.0.2:80;
            proxy_set_header Host $host;
            # Forward client IP
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

```

        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
}

```

در متن بالا به جای Domain.com نام دامنه خود را وارد میکنیم سپس فایل را ذخیره کرده و میندیم پس از آن فرمان زیر را اجرا میکنیم تا nginx با تنظیمات جدید اجرا شود

sudo nginx -s reload

```

ubuntu@test-898892:~$ sudo nginx -s reload
2025/04/07 09:53:10 [notice] 1088403#1088403: signal process started
ubuntu@test-898892:~$

```

تصویر 19-3 راه اندازی مجدد nginx

حالا اگر به آدرس زیر برویم دوباره به صفحه راه اندازی وردپرس میرسیم یعنی دامنه و گواهی HTTPS ما به درستی کار میکند

<https://domain.com/wp-admin>

که domain.com دامنه شما می باشد

در حال حاضر ما سایت خود را اجرا کردیم در حالی که ممکن است این مراحل در نگاه اول سخت به نظر برسد در فصل 4 ما یک راه حل مناسب برای کاهش این مراحل و آسان تر کردن آن داریم در ادامه این فصل به یکسری کارهایی که میتوانیم انجام دهیم و سرعت سایت خود را افزایش دهیم میپردازیم

4-5- فواید اجرای سایت روی سرور به جای خرید سرور هاستینگ

4-5-1- هزینه

یکی از دلایل مهم برای اجرای سایت روی سرور شخصی کاهش هزینه است اگر نصب وردپرس را خودمان روی سرور مد نظر خود انجام دهیم هزینه واسطه گری و هزینه های سربار ناشی از نصب وردپرس و گرفتن SSL یا استفاده از CDN از کل هزینه های ما کاسته می شود جدای از اینکه ما کنترل بیشتری داریم تا منابع مالی خود را بین نیاز های مختلف برای سایت تقسیم کنیم یک نمونه از کاهش هزینه را زیر مشاهده میکنید

سایت [hostgator.com](https://www.hostgator.com) یکی از معروف ترین سایت های هاستینگ است که خدمات هاست سایت های وردپرس را ارائه میکند که تعرفه های هاستینگ این شرکت در عکس زیر نشان داده شده

Hatchling Plan	Baby Plan	Business Plan
Now 66% off!	Now 73% off!	Now 72% off!
\$3.75/mo*	\$4.50/mo*	\$6.25/mo*
Renews at \$10.99/mo	Renews at \$16.49/mo	Renews at \$21.99/mo
Choose Plan	Choose Plan	Choose Plan
10 websites 10GB SSD Storage Chat support Unmetered bandwidth ⓘ Free domain 1st Year ⓘ Pro Email - Free Trial ⓘ Free SSL WordPress pre-installed Malware Scanning	20 websites 20GB SSD Storage Phone & Chat support Unmetered bandwidth ⓘ Free Domain 1st Year ⓘ Pro Email - Free Trial ⓘ Free SSL Wordpress pre-installed Malware Scanning Cloudflare CDN ⓘ Increased performance (2 vCPU's)	50 websites 50GB SSD Storage Phone & Chat support Unmetered bandwidth ⓘ Free Domain 1st Year ⓘ Pro Email - Free Trial ⓘ Free SSL WordPress pre-installed Malware Scanning Cloudflare CDN ⓘ Increased performance (3 vCPU's) Daily Website Back Up - Free 1st Year Domain Privacy - 1st Year

مطابق تصویر بالا برای یک سایت با قابلیت CDN کلودفلر در ماه اول 4.5 دلار و در ماه های بعد هر ماه باید 16.49 دلار هزینه پرداخت شود در حالی که با فرض ثبت نام در سایت هایی که زیرساخت مجانی ارائه میدهند اگر ما خودمان سرور گرفته و دامنه خریداری کنیم تنها هزینه ما هزینه سالانه دامنه خواهد بود که در گران ترین حالت برای دامنه های غیر خاص میشود 15 دلار در سال با یک محاسبه ساده متوجه میشویم که هزینه این کار 8 درصد هزینه خریداری از یک سایت هاستینگ خواهد بود

4-5-2- حریم خصوصی

با انجام تمام مراحل ما کم ترین وابستگی به شرکت طرف سوم برای اجرای سایت خود داریم که امنیت ما را بیشتر و حفظ حریم خصوصی ما را تضمین میکند به خصوص در فضایی که خبر از هک شدن شرکت های بزرگ و لو رفتن داده های مشتریان بیش از قبل به گوش میرسد نمونه یکی از این اتفاقات نشت اطلاعات کاربران سایت [godaddy](https://www.godaddy.com) بود که به لو رفتن اطلاعات کاربران و کلید های SSL منجر شد

4-5-3- به روز بودن

به روز بودن سایت شما در سایت های هاستینگ وابسته به بروز بودن زیرساخت آن شرکت است مثلا ورژن PHP شاید زمانی بعد تر از انتشار نسخه جدید آن در سایت هاستینگ قرار بگیرد در حالی که زمانی که زیرساخت یا همان سرور شما در اختیار شما هست این ارتقا میتواند بدست شما و به ساده ترین روش صورت بگیرد

4-5-4- افزودن قابلیت به پروژه با نرم افزار های متن باز

سایت های هاستینگ در ارائه نرم افزارها برای سایت وردپرسی با محدودیت عمل میکنند در حالی که شما کنترل کامل بر سرور دارید و میتوانید با نرم افزارهای متن باز که مجانی هستند قابلیت های بیشتری به سایت خود اضافه کنید به عنوان مثال با استفاده از فایروال crowdsec میتوانید امنیت سایت خود را بدون هزینه اضافی افزایش دهید

4-5-5- استفاده از ظرفیت آزاد سرور

با در اختیار داشتن سرور شما میتوانید از ظرفیت های باقی مانده از سرور خود برای پروژه های دیگر استفاده کنید یا حتی سایت های بیشتری روی سرور خود میزبانی کنید

4-5-6- افزایش امتیاز عملکرد سایت وردپرس

با در اختیار داشتن سرور شما میتوانید با تغییر تنظیمات سرور مثلا نصب ماژول pagespeed برای nginx امتیاز عملکرد سایت خود را افزایش دهید که به بهبود SEO سایت کمک میکند

4-5-7- محدود بودن به وردپرس

اصولا سایت های هاستینگ سایت به شکل عمده خدمات وردپرس ارائه میدهند در حالی که اگر سایت شما با فریمورک های دیگری طراحی شده باشد نمیتوانید از خدمات آن ها استفاده کنید اما با در کنترل گرفتن سرور شما میتوانید سایت خود را به هر شکلی که طراحی شده باشد میتوانید در فضای اینترنت منتشر کنید

فصل چهارم

جمع بندی

1-5- جمع بندی مراحل قبل

طی این مقاله ما مراحل زیادی بیان کردیم تا سایت خود را نشر دهیم در این بخش ما راه حلی معرفی میکنیم تا تمام مراحل که با اجرای کد در سرور اجرا میشود را در یک مرحله انجام دهیم برای این منظور یک مخزن کد (Repository) آماده شده تا با اجرای یک فرمان کدهای لازم را دانلود و در مکان مناسب قرارداده و فرمان های لازم را اجرا میکند کافی است دامنه و توکن کلودفلر داشته باشید و بقیه کار به شکل خودکار انجام میشود آدرس مخزن کد گیت هاب:

[azad-university-project/setup.bash at main · BIGboss248/azad-university-project](https://github.com/azad-university-project/setup.bash)

2-5- نتیجه گیری و پیشنهادات sss

همانطور که طی این مقاله گفته شد ما با در کنترل گرفتن زیر ساخت (سرور) سایت خود چه ورد پرس یا سایت های دیگری که با ابزار ها و فریم ورک های دیگر طراحی شدند را ارزان تر و سریع تر و امن تر در فضای اینترنت منتشر کنید در نتیجه اگر شما تصمیم به نشر یک سایت دارید میتوانید از مطالب این مقاله استفاده کنید تا سایت خود را بهینه تر منتشر کنید

3-5- منابع و مراجع

منابع ذکر شده در لیست زیر و آدرس های ذکر شده تا تاریخ 22 فروردین 1404 که این منابع لیست شده اند معتبر بوده اند اما این آدرس ها ممکن است در آینده تغییر کند

1- دستور عمل رسمی کار با nginx منتشر شده در سایت رسمی nginx به آدرس [nginx documentation](https://nginx.org/en/docs/)

2- دستور عمل رسمی کار با ورد پرس منتشر شده در سایت رسمی ورد پرس به آدرس [Documentation – WordPress.org](https://wordpress.org/documentation/)

3- دستور عمل رسمی کار با certbot منتشر شده در سایت رسمی certbot به آدرس [Certbot Instructions | Certbot](https://certbot.eff.org/docs/using.html)

4- دستور عمل رسمی استفاده از خدمات Cloudflare منتشر شده در سایت cloudflare به آدرس

[Welcome to Cloudflare | Cloudflare Docs](#)

5- دستور عمل رسمی کار با redis منتشر شده در سایت رسمی Redis به آدرس [Docs](#)

6- دستور عمل کار با docker منتشر شده در سایت رسمی docker به آدرس [Docker Docs](#)

7- دستور عمل کار با vscode منتشر شده در سایت رسمی vscode به آدرس [Documentation for](#)

[Visual Studio Code](#)