Własny model AI na VPS

Pojawienie się takich modeli językowych jak Llama, DeepSeek oraz Bielik spowodowało, że możemy używać LLMów tak jak innego oprogramowania Open Source. Jak w takim razie skonfigurować swój własny model GenAI na serwerze? W tym artykule przejdziemy przez wszystkie niezbędne kroki, używając do tego przede wszystkim Pythona. Napiszemy prostą aplikację webową serwującą odpowiedzi z modelu w FastAPI. Prawie identyczne kroki są aplikowalne do klasycznych modeli uczenia maszynowego, z tą różnicą, że musimy je wcześniej wytrenować sami.

CO JEST WYMAGANE DO PRACY?

Zakładam, że czytelnik jest osobą początkującą, z tego powodu wszystkie kroki opisane są bardzo dokładnie. W razie wątpliwości należy kierować się do repozytorium związanego z artykułem i zadawać pytania w zakładce *Issues* (adres repozytorium: https://github.com/SimonMolinsky/lama-app).

Ważne! Wiele z kroków tworzenia inferencji modeli powtarza się, niezależnie od tego, czy jest to LLM, modele uczenia maszynowego (ML), a nawet proste aplikacje na własnym serwerze. Posiłkując się artykułem, czytelnik może wdrożyć zupełnie inny model. Lektura tego artykułu na pewno rozjaśni też konfigurację inferencji w takich systemach jak AWS SageMaker albo Google Vertex AI.

Modele GenAI i LLM są ciężkie i zajmują wiele pamięci. Nie ma się czemu dziwić, bo przechowują olbrzymie ilości danych tekstowych, które generuje ludzkość. Jednakże dla testów nie warto alokować środków w najlepsze maszyny, które mogą udźwignąć najcięższe modele. My mierzymy w takie, które mogą być uruchomione na VPS ogólnego przeznaczenia (czyli mające mniej mocy GPU i pamięci RAM).. Minimalnie potrzebujemy 8 GB pamięci RAM. Wykorzystamy model Llama3.2 z 3 miliardami parametrów, a programować będziemy w Pythonie. W ostateczności możemy całkowicie pominąć kroki związane z uruchomieniem modelu na VPS (konfiguracja nginx i Gunicorn) i skupić się tylko na krokach, które są wykonywane lokalnie, a treść związaną z konfiguracją serwera potraktować jako przewodnik, kiedy wystąpi taka potrzeba. Tytuły podrozdziałów jasno oznaczają, czy kroki będą wykonywane lokalnie czy na serwerze (VPS).

WSTĘPNA KONFIGURACJA (VPS)

Pierwszym etapem jest konfiguracja serwera i przygotowanie go pod inferencję, czyli odpytywanie wytrenowanego modelu o rekomendacje, predykcje albo prognozy. Patrząc na podstawowe kroki konfiguracji, rodzaj modelu nie ma znaczenia, za to istotne są zasoby sprzętowe. Nie wszystkie modele działają tak samo, nie wszystkie są optymalizowane pod kątem zużycia pamięci VRAM. Z dużą dozą prawdopodobieństwa modele językowe i modele *deep learning* przetwarzające obrazy albo filmy wymagać będą dużych zasobów VRAM, za to modele sekwencyjne, np. złożone systemy prognostyczne dla szeregów czasowych, mogą wymagać dobrych procesorów. Jeszcze inne są modele, które przechowują dużo danych w pamięci RAM, np. niezastąpione *k-NN*. Jako pierwsze implementacje polecam wybierać jak najlżejsze modele, bez zwracania uwagi na ich parametry analityczne i benchmarki. Uczymy się zarządzać infrastrukturą, a nie maksymalizujemy przy tym skuteczności predykcji czy jakości odpowiedzi.

Zanim zaczniemy pisać w terminalu, przejdźmy przez listę kroków, żebyśmy wiedzieli, czego się spodziewać. Zakładam, że konfigurujemy serwer VPS po raz pierwszy. Najpierw trzeba go wykupić, w artykule serwer ma zainstalowany system Ubuntu 24.04 LTS x64. Następnie tworzymy użytkownika i nadajemy mu odpowiednie dostępy, konfigurujemy połączenie między naszym lokalnym komputerem a serwerem. Mając już te kroki za sobą, instalujemy oprogramowanie: Pythona, pip, venv, git. Jeśli chodzi o git, to wysyłamy również publiczny klucz SSH z naszego serwera na GitHuba albo Gitlaba, żeby łączyć się z repozytorium za pomocą SSH, a nie przez HTTPS.

Po przygotowaniu systemu przechodzimy do nieco nużących prac związanych z przygotowaniem struktury katalogów. Warto wydzielić katalogi na aplikację, środowisko wirtualne i logi. Dodatkowo wydziela się katalog na wytrenowany model, ale nie wdrażamy własnego, więc pominiemy ten krok. Ollama pobierze model LLM do specjalnego katalogu na naszym serwerze automatycznie. Podsumowując, są to trzy zadania, które trzeba wykonać zanim zaczniemy myśleć o programowaniu:

1. Zakup VPS i konfiguracja systemu.
2. Instalacja bibliotek i paczek.
3. Przygotowanie struktury projektu.

Korzystanie z zewnętrznego VPS jest o tyle wygodne, że jak coś pójdzie nie tak, to reinstalujemy system i zaczynamy od początku z eksperymentami! Przypominam, że warto zgłaszać wszelkie problemy w repozytorium artykułu, niezależnie czy są to kłopoty z VPS, czy w trakcie pracy we własnym środowisku.

Wybór serwera (VPS)

Wybór serwera rozpoczynamy od wyboru dostawcy. Dostawca musi zapewniać VPS z systemem Ubuntu 24.04 LTS x64 z minimum 8 GB pamięci RAM. Wybieramy też serwer najbliżej naszego miejsca po-

bytu. W sieci jest wiele różnych ofert, najlepiej znaleźć takie, gdzie nie musimy płacić z góry za długi okres użytkowania. To ma być eksperymentalny serwer, a nie produkcyjna maszyna. Najlepiej zrobić wcześniej mały research na Reddit. Ja używam DigitalOcean, ale nie jest to najtańsza opcja na rynku (dalej jest zestawienie kilku dostawców, możemy tam zerknąć i ich porównać). Wszystkie kroki z artykułu można wykonać w ciągu godziny zegarowej, więc taki koszt nie jest w tej sytuacji przerażający, ale trzeba pamiętać o usunięciu serwera VPS po zakończeniu pracy, bo jeśli tego nie zrobimy, będziemy płacić za każdą godzinę alokacji infrastruktury!

Przykładowi dostawcy i ceny za VPS z 8 GB RAM to:

» Digital Ocean – 48 USD/mc, 0,071 USD/godzinę.
» Hetzner – 7.59 USD/mc, 0,0127 USD/godzinę.
» Linode – 48 USD/mc, 0,072 USD/godzinę.

Od tych dostawców możemy rozpocząć poszukiwania. Pamiętajmy też o systemie operacyjnym, na którym będziemy pracować (Ubuntu 24.04 LTS x64).

Uwaga! Czasami przy konfiguracji VPS jesteśmy proszeni o dostarczenie kluczy SSH. Jeśli taka sytuacja wystąpi, zróbmy to! Jak wygenerować klucz SSH? Zobacz rozdział *Konfiguracja kluczy SSH*, gdzie opisano, jak to zrobić dla systemów UNIXowych.

Pierwsze połączenie z serwerem (VPS)

Jeśli dostawca serwera wymagał wcześniejszego wygenerowania kluczy SSH, to będzie dużo wygodniej i po połączeniu z serwerem możemy przejść do kroku *Utworzenie dedykowanego użytkownika*.

Połączenie z serwerem następuje z poziomu terminala (Listing 1).

Listing 1. Połączenie SSH z serwerem

```
ssh nazwa-uzytkownika@ip-servera
```

Tekst `nazwa-uzytkownika` to najczęściej `root`, ale niektórzy dostawcy usług nie pozwalają na logowanie się na konto administratora (`root`) i podają własnego użytkownika, np. `ubuntu`. Adres IP powinien przyjść w emailu albo pojawić się w panelu administracji serwera. Może to być na przykład 167.170.140.30. Jeśli klucze SSH są już na serwerze, to połączenie (po potwierdzeniu chęci połączenia do nowego hosta) powinno pójść bez problemu. Czasami musimy podać jeszcze hasło, ale ono jest zawsze wysyłane przez dostawcę.

Konfiguracja kluczy SSH (VPS)

Niektórzy dostawcy nie pozwalają na ustawienie kluczy uwierzytelniających na etapie wykupowania serwera. Wtedy musimy zrobić to już po pierwszym połączeniu (zapewne przez nazwę użytkownika i hasło).

1. Najpierw otwórzmy terminal na swoim komputerze i wygenerujmy parę kluczy, używając polecenia `ssh-keygen`. Jeśli pojawi się pytanie, czy nadpisać klucz, bo on już istnieje, wtedy należy zaprzeczyć! Skorzystamy z istniejącego klucza i nie będziemy go nadpisywać, bo może być gdzieś używany.
2. Zostaniemy zapytani o nazwę klucza, pozostawmy ją taką, jaka jest, czyli id_rsa albo id_ed25519. Klucz zostanie zapisany do katalogu /home/nazwa-uzytkownika/.ssh/.
3. Jeśli tworzymy nowy klucz, zostaniemy zapytani o *passphrase*, czyli dodatkowy poziom zabezpieczeń, który chroni nasz prywatny klucz przed wykradzeniem nawet wtedy, gdy ktoś inny ma dostęp do naszego komputera. Jest to opcjonalny, ale zalecany krok. Passphrase podajemy przy każdym nowym połączeniu SSH do zewnętrznego serwisu, tak jakbyśmy wpisywali hasło. Możemy też trzymać passphrase w *cache*, żeby nie wpisywać jej za każdym razem.
4. Klucz jest utworzony!
5. Jeśli o klucz pyta dostawca usług serwerowych w oknie instalacji VPS, przekopiujmy go z pliku */home/nazwa-uzytkownika/.ssh/id_rsa.pub* (lub id_ed25519.pub). Uwaga! Plik musi mieć końcówkę `.pub`! Jest to klucz publiczny, który można udostępniać, w przeciwieństwie do zawartości pliku bez końcówki `.pub`, który zawiera klucz prywatny, którego nie wolno udostępniać. Jeśli to zrobimy, wtedy nasz klucz nie jest już bezpieczny i musimy wygenerować nowy. Najwygodniej jest otworzyć ten plik od razu w terminalu i skopiować jego treść ręcznie: `cat ~/.ssh/id_rsa.pub` albo `cat ~/.ssh/id_ed25519.pub`
6. W terminalu pojawi się ciąg znaków zaczynający się od frazy `ssh-rsa` albo `ssh-ed25519`. Skopiujmy go, łącznie z tą frazą, i przeklejmy do okna dostawy usług serwerowych.
7. W przypadku podłączania klucza samemu, w terminalu wpiszmy kod z Listingu 2.

Listing 2. Przekopiowanie klucza SSH na serwer

```
ssh-copy-id nazwa-uzytkownika@ip-servera
```

8. Teraz możemy łączyć się bezpiecznie z serwerem! Mając skonfigurowane klucze uwierzytelniające, możemy też zrezygnować z wpisywania hasła przy każdym połączeniu, ale nie będziemy tego tematu rozwijać w tym artykule.

Utworzenie dedykowanego użytkownika (VPS)

Pierwsze połączenie do serwera VPS prowadzi do profilu administracyjnego (`root` albo inny profil z wystarczająco szerokimi uprawnieniami). Dobrą praktyką jest utworzenie profilu dla aplikacji – w naszym przypadku profilu do serwowania inferencji z modelu uczenia maszynowego.

1. Połączmy się z serwerem (zakładamy, że główny użytkownik to `root`): `ssh root@ip-serwera`.
2. Utworzymy użytkownika, ja nazwę go `lama`, w terminalu będzie trzeba odpowiedzieć na kilka pytań. Najważniejsze to podanie hasła, resztę można pominąć. Użyjmy polecenia: `adduser lama`.
3. Użytkownik `lama` dostanie szerokie uprawnienia, więc w terminalu wpisujemy: `usermod -aG sudo lama`.
4. Teraz skopiujemy klucz SSH z naszego głównego konta do konta nowo utworzonego użytkownika (Listing 3).

Listing 3. Kopia kluczy SSH z konta administracyjnego na konto użytkownika

```
rsync --archive --chown=lama:lama ~/.ssh /home/lama
```

5. Otwórzmy nowy terminal na swoim komputerze i wpiszmy: `ssh lama@ip-serwera`.

PAIR PROGRAMMING – CZY TO DZIAŁA? · STREFA CVE

INDEX: 285358

www.programistamag.pl

Magazyn programistów i liderów zespołów IT

programista

2/2025 (117)

Cena 32.90 zł (w tym VAT 8%)

CZY „CZYSTY KOD" WYTRZYMAŁ PRÓBĘ CZASU?

ISSN 2084-9400

02

9 772084 940503

JAK DZIAŁA INTERNET - TCP I NAT

WŁASNY [illegible] NA VPS

MEGA65 - WSKRZESZONY NASTĘPCA WIELKIEJ LEGENDY

[illegible]ENA AUTOMATYZACJA [illegible]EN DLA ASP.NET

NOWY NUMER JUZ W EMPIKACH