

**Academia de Studii Economice, București**

**Facultatea de Cibernetică, Statistică și Informatică Economică**

**Specializarea: Informatică Economică**

DevOps pentru Inteligența Artificială

**Student**: Soare Florin-Răzvan

**Student**: Sturza Mihai-George

**București**

**2023**

***Curpins***

[**1.** **Introducere în DevOps** 3](#_Toc155529552)

[**2.** **Practici și Metodologii în DevOps pentru Inteligența Artificială** 4](#_Toc155529553)

[2.1 Integrarea Continuă (CI) 4](#_Toc155529554)

[2.2 Livrarea Continuă (CD) 5](#_Toc155529555)

[2.3 Automatizarea Infrastructurii și Configurației 5](#_Toc155529556)

[2.4 Monitorizarea și Feedback-ul Continuu 5](#_Toc155529557)

[**3.** **Unelte și Tehnologii pentru Deploying Modele de Inteligență Artificială** 6](#_Toc155529558)

[3.1 Deploying pe Edge Devices (Telefoane, Camere, etc.) 6](#_Toc155529559)

[3.2 Deploying pe Physical Servers 6](#_Toc155529560)

[3.3 Developying în Cloud 6](#_Toc155529561)

[**4.** **Ciclul de viață al modelelor de Inteligență Artificială** 7](#_Toc155529562)

[4.1 Antrenarea 7](#_Toc155529563)

[a. Antrenarea pe Edge Devices 7](#_Toc155529564)

[b. Antrenarea pe Servere Fizice 7](#_Toc155529565)

[c. Antrenarea în Cloud 7](#_Toc155529566)

[4.2 Stocarea 8](#_Toc155529567)

[a. Stocarea pe Edge Devices 8](#_Toc155529568)

[b. Stocarea pe Servere Fizice 8](#_Toc155529569)

[c. Stocarea în Cloud 8](#_Toc155529570)

[4.3 Inferențe 9](#_Toc155529571)

[a. Inferența pe Edge Devices(Telefoane,Camere, etc.) 9](#_Toc155529572)

[b. Inferența pe Physical Servers 9](#_Toc155529573)

[c. Inferența în Cloud 9](#_Toc155529574)

[4.3.1 Crearea API-urilor pentru Servere și Cloud 10](#_Toc155529575)

[**5.** **Moduri de servire ale modelelor de Inteligență Artificială** 10](#_Toc155529576)

[5.1 Web APIs 10](#_Toc155529577)

[5.2 Tensorflow.js 12](#_Toc155529578)

[**6.** **Beneficii și Provocări în Implementarea DevOps pentru Inteligența Artificială** 12](#_Toc155529579)

[6.1 Îmbunătățirea Calității Software-ului 12](#_Toc155529580)

[a. Beneficii 12](#_Toc155529581)

[b. Provocări 13](#_Toc155529582)

[6.2 Viteza și Eficiența în Livrare 13](#_Toc155529583)

[a. Beneficii 13](#_Toc155529584)

[b. Provocări 13](#_Toc155529585)

[6.3 Provocări comune 13](#_Toc155529586)

[**Bibliografie** 14](#_Toc155529587)

1. **Introducere în DevOps**

Termenul "DevOps" este format din cuvintele "development" (dezvoltare) și "operations" (operațiuni). DevOps reprezintă o serie de practici și filosofii care au ca scop îmbunătățirea colaborării și eficienței dintre echipele de dezvoltare software și cele de operațiuni IT. Obiectivul principal este de a integra și automatiza procesele între aceste departamente pentru a facilita construirea, testarea și lansarea software-ului într-un mod mai rapid și mai eficient. DevOps implică modificări în modul în care echipele IT colaborează în cadrul ciclului de viață al dezvoltării software-ului. Acest lucru ajută la reducerea timpilor de lansare pe piață, la îmbunătățirea calității software-ului și la creșterea satisfacției clienților.

Originea DevOps poate fi urmărită înapoi la începutul anilor 2000, când industria IT a început să conștientizeze nevoia de o mai bună colaborare și comunicare între dezvoltatori și profesioniștii IT responsabili de operațiuni. Această nevoie a fost amplificată de evoluția rapidă a tehnologiilor și de cererea crescândă pentru software de înaltă calitate livrat într-un ritm accelerat.

Ideea de DevOps a prins contur în mod formal în 2009, când Patrick Debois, un inginer belgian, a organizat prima conferință "DevOpsDays" la Gent, în Belgia. Evenimentul a reunit profesioniști din domeniul dezvoltării și operațiunilor pentru a discuta despre modalități de îmbunătățire a integrării între aceste două domenii. Debois este adesea menționat ca fondatorul mișcării DevOps.

A person speaking into a microphone

Description automatically generated

Figure 1: Patrick Debois

Creșterea DevOps a fost de asemenea influențată de popularizarea metodologiilor Agile de dezvoltare software. Agile pune accent pe adaptabilitate, livrare incrementală și colaborare strânsă între echipe, valorile care se aliniază strâns cu principiile DevOps. În timp, DevOps a evoluat de la a fi o idee de nișă la a deveni o practică standard în industria IT. Acest lucru a fost posibil datorită recunoașterii valorii sale în accelerarea ciclului de dezvoltare a produselor software, îmbunătățirea calității și eficienței operaționale și facilitarea unei mai bune colaborări între echipele de dezvoltare și cele de operațiuni.

Astăzi, DevOps este văzut ca un element esențial în cadrul organizațiilor care doresc să fie agile și competitive în peisajul digital în continuă schimbare. Cu ajutorul unor unelte și tehnologii inovatoare, împreună cu adoptarea unei culturi organizaționale axate pe colaborare și îmbunătățire continuă, DevOps continuă să modeleze modul în care software-ul este dezvoltat și livrat în era digitală.

1. **Practici și Metodologii în DevOps pentru Inteligența Artificială**
   1. Integrarea Continuă (CI)

Integrarea Continuă în contextul Inteligenței Artificiale (AI) este crucială. Aceasta implică integrarea automată a codului și a modelelor AI într-un repository comun, asigurându-se că schimbările sunt compatibile și funcționale. În AI, CI trebuie să includă nu doar integrarea codului, ci și a datelor, modelelor și algoritmilor. Validarea continuă a modelului AI prin teste automate și evaluarea performanței sunt esențiale pentru a asigura fiabilitatea și eficacitatea soluțiilor AI. Aceasta reduce semnificativ timpul necesar pentru a identifica și corecta erori, contribuind la accelerarea ciclului de dezvoltare.

* 1. Livrarea Continuă (CD)

Livrarea Continuă în domeniul AI se referă la automatizarea procesului de lansare a modelelor AI în producție. Scopul este de a facilita o implementare rapidă și eficientă a modelelor AI, după ce acestea au fost testate și validate în etapa de CI. CD în AI trebuie să gestioneze provocări specifice, cum ar fi gestionarea versiunilor datelor, actualizarea modelelor și asigurarea conformității cu standardele de securitate și confidențialitate. Această etapă include și pregătirea infrastructurii necesare pentru a rula modelele AI la scală, menținând performanța și fiabilitatea.

* 1. Automatizarea Infrastructurii și Configurației

Automatizarea infrastructurii și configurației în contextul AI este deosebit de importantă datorită cerințelor specifice de calcul și stocare. Automatizarea trebuie să asigure că resursele necesare sunt disponibile și optimizate pentru a rula modelele AI, inclusiv servere cu capacitate mare de procesare și stocare, precum și hardware specifice (de exemplu, GPU-uri pentru antrenarea modelelor de învățare automată). De asemenea, automatizarea trebuie să se ocupe de configurarea și actualizarea mediilor de dezvoltare și testare pentru AI, asigurând coerența și reducerea erorilor umane.

* 1. Monitorizarea și Feedback-ul Continuu

Monitorizarea și feedback-ul continuu sunt vitale în ciclul de viață al soluțiilor AI. Aceasta include monitorizarea performanței modelelor AI în producție, detectarea timpurie a abaterilor sau erorilor, și colectarea feedback-ului pentru îmbunătățirea continuă. În AI, este esențial să se monitorizeze nu doar performanța tehnică, ci și acuratețea și biasul modelului, pentru a asigura rezultate fiabile și etice. Feedback-ul continuu din partea utilizatorilor finali și a stakeholder-ilor este, de asemenea, crucial pentru ajustarea și îmbunătățirea modelelor AI.

1. **Unelte și Tehnologii pentru Deploying Modele de Inteligență Artificială**
   1. Deploying pe Edge Devices (Telefoane, Camere, etc.)

Deploying-ul modelelor de inteligență artificială (AI) pe edge devices, cum ar fi telefoanele inteligente și camerele de supraveghere, este o practică tot mai răspândită. Aceste dispozitive au devenit suficient de puternice pentru a rula modele AI complexe, oferind avantajul procesării datelor direct la sursă. Acest lucru reduce latența și dependența de conectivitatea la internet, fiind ideal pentru aplicații în timp real, cum ar fi recunoașterea facială, asistenți virtuali personali și monitorizarea de securitate.

Pentru deploying pe edge devices, este crucială optimizarea modelului AI pentru a rula eficient pe hardware cu resurse limitate. Se utilizează tehnici precum pruning, quantization și model distillation pentru a reduce dimensiunea și complexitatea modelului, fără a sacrifica semnificativ acuratețea.

* 1. Deploying pe Physical Servers

Deploying modele AI pe servere fizice oferă control complet asupra resurselor hardware și software. Acest lucru este adesea preferat pentru aplicații care necesită un grad ridicat de personalizare, securitate sau pentru cazurile în care performanța și predictibilitatea sunt critice. Serverele fizice permit configurarea detaliată a mediului de operare, inclusiv alegerea specifică a CPU-urilor, GPU-urilor și a altor resurse hardware.

Pentru a maximiza eficiența acestor sisteme, se utilizează tehnologii avansate de virtualizare și containerizare, cum ar fi Docker și Kubernetes, care permit o gestionare flexibilă și scalabilă a resurselor și a aplicațiilor.

* 1. Developying în Cloud

Cloud computing-ul a devenit un element central în deploying-ul modelelor AI, oferind flexibilitate, scalabilitate și acces la hardware de ultimă generație. Furnizorii de servicii cloud, cum ar fi AWS, Google Cloud Platform și Microsoft Azure, oferă servicii specializate pentru AI, inclusiv infrastructuri optimizate pentru învățare automată și învățare profundă.

Avantajele deploying-ului în cloud includ accesul la resurse computaționale de înaltă performanță (cum ar fi GPU-uri și TPU-uri), scalabilitatea rapidă în funcție de cerințele de procesare și stocare, precum și servicii de gestionare și monitorizare integrate. De asemenea, cloud-ul facilitează colaborarea și accesul la date și modele între echipele distribuite geografic.

1. **Ciclul de viață al modelelor de Inteligență Artificială**
   1. Antrenarea
2. Antrenarea pe Edge Devices

Antrenarea modelelor de AI direct pe edge devices, cum ar fi telefoanele și camerele, este adesea limitată datorită resurselor hardware restrânse. Cu toate acestea, progresele recente în hardware-ul mobil permit implementarea unor tehnici simple de învățare automată. De exemplu, tehnologii precum TensorFlow Lite și Core ML permit rularea algoritmilor de AI pe dispozitive mobile, oferind un grad de învățare și adaptare în timp real.

1. Antrenarea pe Servere Fizice

Serverele fizice sunt ideale pentru antrenarea modelelor AI complexe, oferind resurse computaționale extinse și control complet asupra mediului de antrenare. Aceste servere pot fi echipate cu hardware de înaltă performanță, cum ar fi GPU-uri și CPU-uri puternice, pentru a accelera procesul de antrenare. În plus, serverele fizice permit stocarea și procesarea unor cantități mari de date, esențiale pentru antrenarea modelelor AI.

1. Antrenarea în Cloud

Cloud computing-ul este o alegere populară pentru antrenarea modelelor AI datorită scalabilității și flexibilității sale. Furnizorii de cloud oferă acces la infrastructuri puternice și specializate, cum ar fi GPU-uri și TPU-uri, care pot procesa seturi mari de date și pot antrena modele complexe eficient. De asemenea, cloud-ul facilitează experimentarea și iterația rapidă prin provisionarea rapidă a resurselor.

* 1. Stocarea

1. Stocarea pe Edge Devices

Stocarea modelelor AI pe edge devices necesită optimizarea mărimii modelului pentru a se potrivi cu capacitatea limitată de stocare. Modelele sunt adesea comprimate și optimizate înainte de a fi încărcate pe dispozitiv, folosind tehnici precum pruning-ul și cuantizarea pentru a reduce dimensiunea fără a compromite semnificativ performanța.

1. Stocarea pe Servere Fizice

Serverele fizice oferă spațiu de stocare semnificativ, permițând păstrarea unor seturi de date mari și a multiple versiuni ale modelelor AI. De asemenea, ele permit implementarea unor soluții de backup și de recuperare a datelor pentru a asigura integritatea și securitatea informațiilor.

1. Stocarea în Cloud

Cloud-ul oferă opțiuni versatile și scalabile pentru stocarea modelelor AI și a datelor de antrenare. Serviciile de stocare în cloud pot scala pentru a se potrivi cu dimensiunea și cerințele specifice ale proiectului, oferind în același timp un nivel ridicat de securitate și redundanță. De asemenea, cloud-ul permite accesul facil la date și modele de oriunde, facilitând colaborarea între echipele distribuite geografic.

* 1. Inferențe

1. Inferența pe Edge Devices(Telefoane,Camere, etc.)

Inferența AI pe edge devices se concentrează pe procesarea datelor direct la sursă, ceea ce este esențial pentru aplicații care necesită reacții rapide și autonomie de la conexiunea de rețea.

* **Telefoane inteligente**: Acestea pot efectua inferență pentru funcții precum recunoașterea vocală, traduceri în timp real, și alte aplicații AI integrate. Avantajul principal este capacitatea de a procesa datele local, reducând astfel latența și utilizarea datelor mobile.
* **Camere de supraveghere**: Camerele inteligente pot efectua inferență pentru recunoașterea facială, detectarea mișcărilor sau analize de securitate. Procesarea locală permite acțiuni rapide și eficiente fără întârzieri semnificative.

1. Inferența pe Physical Servers

Inferența pe servere fizice este ideală pentru sarcini care necesită o capacitate de procesare mare și o securitate sporită.

* **Aplicații de Business Critice**: Serverele fizice sunt folosite pentru aplicații unde viteza de procesare și fiabilitatea sunt esențiale, cum ar fi în sisteme bancare sau de monitorizare a infrastructurilor critice.
* **Centre de Date**: În centrele de date, inferența AI se poate efectua la o scară mare, gestionând simultan multiple fluxuri de date și asigurând o procesare rapidă și eficientă.

1. Inferența în Cloud

Inferența AI în cloud este ideală pentru flexibilitate, scalabilitate și acces la resurse computaționale avansate.

* **Scalabilitate și Flexibilitate**: Cloud-ul oferă capacitatea de a scala resursele în funcție de cerere, fiind ideal pentru aplicații cu fluctuații mari în utilizare.
* **Servicii Diverse**: Furnizorii de cloud oferă o gamă largă de servicii AI gata de utilizat, cum ar fi recunoașterea vocală, procesarea limbajului natural și analiza imaginilor, care pot fi integrate rapid în aplicații.
  + 1. Crearea API-urilor pentru Servere și Cloud

Pentru mediile de servere fizice și cloud, există posibilitatea de a crea API-uri (Interfețe de Programare a Aplicațiilor) care facilitează accesul la modelele AI pentru diferite aplicații și servicii. API-urile permit integrarea ușoară a inferenței AI în aplicații diverse, oferind o modalitate standardizată și securizată de a accesa și utiliza modelele AI stocate pe servere sau în cloud. Aceasta include posibilitatea de a procesa cereri de inferență în timp real, de a gestiona cererile de la multiple aplicații și de a oferi rezultatele într-un format ușor de utilizat. Utilizarea API-urilor este fundamentală pentru a oferi accesibilitate și scalabilitate serviciilor de AI, făcându-le disponibile pentru o gamă largă de aplicații și utilizatori.

1. **Moduri de servire ale modelelor de Inteligență Artificială**
   1. Web APIs

Web APIs (Interfețe de Programare a Aplicațiilor) sunt esențiale pentru a servi modele de Inteligență Artificială (AI) către diverse aplicații și platforme. Ele funcționează prin primirea și procesarea cererilor HTTP, oferind o modalitate standardizată și eficientă de a interacționa cu funcționalitățile AI.

Caracteristici principale:

* **Rulare pe Client**: Execută modele AI direct în browser, reducând necesitatea de a trimite datele pe un server.
* **Interactivitate**: Ideal pentru aplicații web interactive, cum ar fi recunoașterea vizuală și audio în timp real.
* **Independență de Platformă**: Fiind bazat pe JavaScript, este compatibil cu majoritatea browserelor moderne și poate fi integrat ușor în aplicații web existente.
* **Protecția Confidențialității**: Prin procesarea datelor pe dispozitivul clientului, se reduce riscul de expunere a datelor sensibile.

Un request HTTP tipic către un Web API pentru AI include următoarele componente:

* URL (Uniform Resource Locator):

Identifică adresa Web API-ului și include baza URL-ului, calea resursei și parametrii de interogare.

* Metoda HTTP:

Definește tipul acțiunii dorite(de exemplu, GET pentru solicitarea datelor, POST pentru trimiterea datelor către server); pentru modele AI, POST este adesea utilizat pentru a trimite datele de intrare pentru inferență.

* Headere HTTP:

Conțin informații suplimentare despre request, cum ar fi tipul de conținut (Content-Type), autorizarea și alte metadate. De exemplu, pentru un request care trimite date JSON, header-ul **Content-Type** ar fi setat la **application/json**.

* Corpul Request-ului:

În cazul metodei POST, corpul request-ului conține datele efective care trebuie procesate de API (de exemplu, imaginea sau textul pentru inferență AI). Datele pot fi în diferite formate, cum ar fi JSON, form-data sau chiar binar, în funcție de API și de tipul datelor trimise.

* Parametrii de interogare:

Opțional, pot fi adăugați la URL pentru a specifica anumite detalii ale request-ului (de exemplu, parametrii pentru configurarea inferenței).

* Răspunsul Serverului:

După procesarea request-ului, serverul răspunde cu un cod de status HTTP (de exemplu, 200 pentru succes, 400 pentru request greșit) și, de obicei, un corp de răspuns. Corpul răspunsului poate conține rezultatul inferenței AI, precum și orice mesaje sau metadate relevante.

* 1. Tensorflow.js

Tensorflow.js este o bibliotecă JavaScript dezvoltată de Google, care permite antrenarea și deploy-ul modelelor de AI direct în browser sau în mediu Node.js. Această abordare aduce inteligența artificială mai aproape de utilizatorul final, permițând rularea inferenței AI direct pe dispozitivul clientului.

Caracteristici principale:

* **Rulare pe Client**: Execută modele AI direct în browser, reducând necesitatea de a trimite datele pe un server.
* **Interactivitate**: Ideal pentru aplicații web interactive, cum ar fi recunoașterea vizuală și audio în timp real.
* **Independență de Platformă**: Fiind bazat pe JavaScript, este compatibil cu majoritatea browserelor moderne și poate fi integrat ușor în aplicații web existente.
* **Protecția Confidențialității**: Prin procesarea datelor pe dispozitivul clientului, se reduce riscul de expunere a datelor sensibile.

1. **Beneficii și Provocări în Implementarea DevOps pentru Inteligența Artificială**
   1. Îmbunătățirea Calității Software-ului
2. Beneficii

* **Integrare și Testare Continuă**: Prin DevOps, echipele pot integra și testa codul în mod continuu, detectând și corectând rapid erorile. Aceasta duce la îmbunătățirea calității generale a software-ului și a modelelor de AI.
* **Feedback Rapid**: Ciclurile scurte de feedback permit ajustări rapide și îmbunătățirea continuă, asigurând că software-ul răspunde eficient nevoilor utilizatorilor și cerințelor pieței.
* **Colaborare Îmbunătățită**: DevOps încurajează colaborarea strânsă între dezvoltatori, operatori și alte părți interesate, ceea ce duce la o mai bună aliniere și înțelegere a obiectivelor și cerințelor proiectului.

1. Provocări

* **Complexitatea în Managementul Datelor**: Gestionarea datelor pentru AI poate fi complexă, mai ales în ceea ce privește calitatea, volumul și securitatea datelor.
* **Asigurarea Calității Modelelor AI**: Este provocator să se asigure că modelele de AI sunt precise, imparțiale și robuste, ceea ce necesită procese riguroase de testare și validare.
  1. Viteza și Eficiența în Livrare

1. Beneficii

* **Automatizare Extinsă**: Automatizarea proceselor de dezvoltare și operațiuni prin DevOps conduce la reducerea timpilor de livrare și îmbunătățirea eficienței.
* **Scalabilitate și Flexibilitate**: DevOps oferă capacitatea de a scala și adapta rapid resursele în funcție de cerințele proiectului, ceea ce este vital în proiectele de AI.
* **Livrare Continuă**: Capacitatea de a implementa în mod continuu îmbunătățiri și actualizări asigură că software-ul rămâne relevant și competitiv.

1. Provocări

* **Gestionarea Resurselor și a Costurilor**: Scalarea resurselor, mai ales în cloud, poate crește semnificativ costurile, necesitând o gestionare atentă a bugetului.
* **Complexitatea Infrastructurii**: Infrastructurile necesare pentru AI, în special în cloud, pot fi complexe și greu de gestionat fără expertiza adecvată.
  1. Provocări comune
* **Lipsa de Expertiză**: Expertiza în DevOps și AI poate fi rară, ceea ce face dificilă construirea și menținerea unei echipe eficiente.
* **Securitatea și Conformitatea**: Asigurarea securității datelor și a modelelor de AI, precum și conformitatea cu reglementările privind protecția datelor, pot fi provocatoare.
* **Gestionarea Schimbărilor**: Implementarea unei culturi DevOps necesită schimbări semnificative în modul de lucru și poate întâmpina rezistență în cadrul organizației.
* **Integrarea Uneltelor și Tehnologiilor**: Alegerea și integrarea uneltelor potrivite pentru DevOps și AI poate fi dificilă, datorită diversității și specificităților tehnologice.
* **Gestionarea Dependențelor și a Versiunilor**: În AI, gestionarea dependențelor, a seturilor de date și a versiunilor modelelor poate fi complexă.
* **Evaluarea Performanței și Acurateței**: Măsurarea și evaluarea performanței și acurateței modelelor de AI în condiții reale de operare reprezintă un alt obstacol comun.

# **Bibliografie**

1. <https://en.wikipedia.org/wiki/DevOps>
2. <https://about.gitlab.com/topics/devops/>
3. <https://www.atlassian.com/devops>
4. <https://devops.com/the-origins-of-devops-whats-in-a-name/>
5. <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-devops/>
6. <https://www.knowledgehut.com/blog/devops/history-of-devops>
7. <https://www.techtarget.com/whatis/reference/The-history-of-DevOps-A-visual-timeline>