



UNIVERSITATEA DIN
BUCUREȘTI

FACULTATEA DE
MATEMATICĂ ȘI
INFORMATICĂ



SPECIALIZAREA TEHNOLOGIA INFORMAȚIEI

Lucrare de licență

DEVOPS ÎN MICROSERVICII

Absolvent

Bahrim Dragoș

Coordonator științific

Conferențiar Doctor Kevorchian Cristian

București, iulie 2023

Rezumat

Microserviciile sunt o arhitectură de sisteme distribuite ce a luat amploare în urma structurării organizațiilor către piață. Aceasta a fost facilitată de schimbări în modul în care lansăm produsele prin dezvoltarea platformelor cloud și a ce a dus la îmbunătățiri la timpul de livrare dar și a consistenței. Obiectivul este prezentarea acestor concepte și implementarea acestora prezentând modul de gândire ce influențează ciclul produsului. În acest scop, mă documentez legat de domeniu și încerc să îmi fac o idee de ansamblu asupra aspectelor ce trebuie luate în construirea unui sistem iar apoi o să aplic aceste informații. Aceasta lucrare ar trebui să servească ca un prim contact cu microserviciile dar și a practicilor DevOps.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce vitae eros sit amet sem ornare varius. Duis eget felis eget risus posuere luctus. Integer odio metus, eleifend at nunc vitae, rutrum fermentum leo. Quisque rutrum vitae risus nec porta. Nunc eu orci euismod, ornare risus at, accumsan augue. Ut tincidunt pharetra convallis. Maecenas ut pretium ex. Morbi tellus dui, viverra quis augue at, tincidunt hendrerit orci. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam quis sollicitudin nunc. Sed sollicitudin purus dapibus mi fringilla, nec tincidunt nunc eleifend. Nam ut molestie erat. Integer eros dolor, viverra quis massa at, auctor.

Cuprins

| | | |
|----------|---------------------------------------|-----------|
| 1 | Introducere | 5 |
| 2 | Preliminarii | 8 |
| 3 | Microservicii | 9 |
| 3.1 | Concepte generale | 9 |
| 3.2 | Design | 9 |
| 3.3 | Tehnici de implementare | 9 |
| 3.4 | Build | 9 |
| 3.5 | Testare | 9 |
| 3.6 | Deployment | 9 |
| 3.7 | Monitorizare | 9 |
| 3.8 | Securizare | 9 |
| 3.9 | Evoluție | 9 |
| 4 | Orchestrare | 10 |
| 4.1 | Istoric deployment | 10 |
| 4.2 | Tipuri de deployment | 10 |
| 4.2.1 | Baremetal | 10 |
| 4.2.2 | Mașină virtuală | 10 |
| 4.2.3 | Container | 10 |
| 4.2.4 | Function as a Service | 10 |
| 4.2.5 | Platform as a Service | 10 |
| 4.2.6 | Container as a Service | 10 |
| 4.3 | Docker | 10 |
| 4.4 | Docker Swarm | 10 |
| 4.5 | Kubernetes | 10 |
| 4.6 | Red Hat OpenShift | 10 |
| 4.7 | HashiCorp Nomad | 10 |
| 4.8 | Infrastructure as a Service | 10 |

| | | |
|----------|---|-----------|
| 5 | Îmbunătățirea metodelor de dezvoltare software | 11 |
| 5.1 | Schimbari in livrarea produselor | 11 |
| 5.2 | DevOps și aspecte ale comunității | 11 |
| 5.3 | Ce înseamnă să aplici DevOps | 11 |
| 5.4 | Unelte | 11 |
| 6 | Aplicație | 12 |
| 7 | Concluzii | 13 |

Capitolul 1

Introducere

Domeniul informaticii, calculatoarelor și al tehnologiei informației este unul care va fi mereu în continuă evoluție. Dorința companiilor de a-și servi clienții cât mai rapid a dus la un avans tehnologic în care mereu apare ceva nou ce are ca scop îmbunătățirea proceselor actuale. Această arie și-a început dezvoltarea aproape acum o sută de ani iar avansul poate fi apreciat numai gândindu-ne la modul în care scriam cod și livram produse acum, acum zece ani și acum două zeci de ani. Se observă o diferență și la accesibilitatea unităților de calcul, actual majoritatea persoanelor au acces la un dispozitiv ce se poate conecta la Internet, ceea ce le permite să devină consumatori la diferite servicii.

Piața pentru unități de calcul a evoluat. În 1943, Thomas Watson, director la IBM menționează că „I think there is a world market for maybe five computers” (Cred că există o piață globală pentru poate cinci computere). În 1999, Michael Barr afirmă în „Programming Embedded Systems in C and C++” „One of the more surprising developments of the last few decades has been the ascendance of computers to a position of prevalence in human affairs. Today there are more computers in our homes and offices than there are people who live and work in them. Yet many of these computers are not recognized as such by their users.” (Una dintre cele mai surprinzătoare dezvoltări a ultimelor decenii a fost ascensiunea computerelor într-o poziție predominantă în afacerile oamenilor. Astăzi sunt mai multe computere în casele noastre și în birouri decât persoane care trăiesc și lucrează în ele. Însă multe dintre acestea nu sunt recunoscute de către utilizatorii lor). Cele două fraze evidențiază modul cum s-a schimbat prezența computerelor în viața noastră. La început acestea erau foarte costisitoare, greu de administrat și de operat însă acestea au devenit din ce în ce mai mici iar recent aproape orice are nevoie de semiconductori datorită integrării ce le fac „smart”.

Aceste afirmații se aplică și pentru produsele software, ce au ca menire să ofere asistență altor aplicații sau să fie consumate direct. Indiferent de locul în care mă duc, probabil persoana cu care aş interacționa folosește un computer, fie că îmi cumpăr ceva de la un magazin făcând o plată cu cardul sau că mi se livrează un colet iar curierul marchează pe AWB că a fost livrat. Christopher Little afirmă că „Every company is a technology com-

pany, regardless of what business they think they are in. A bank is just an IT company with a banking license.” (Orice companie este o companie ce se axează pe tehnologie, indiferent de mediul de afaceri în care se află. O bancă este doar o companie IT cu licență de a funcționa ca o bancă.)

Desigur nu toate companiile sunt la fel și nu toate concurează cu toate companiile, însă chiar dacă nu ești o companie ce activează în tehnologie, probabil folosești componente tehnologice pentru a-ți îmbunătăți randamentul sau să oferi clienților un avantaj față de competiție, care probabil gândește în același mod.

Însă pentru fiecare companie obiectivul este identic, să livreze clienților produse de calitate și cât mai rapid. Pentru aceasta trebuie să ne asigurăm că avem inginerii necesari pentru implementarea cererilor noi, însă momentul în care aceasta este terminată este doar începutul procesului de integrare și ulterior lansare, și uneori acesta este cel care reprezintă un blocaj din mai multe puncte de vedere ce limitează numărul de câte ori putem aduce o îmbunătățire produsului nostru.

În încercarea de a livra mai rapid sau pentru a ușura modul în care manevrăm un influx sau o lipsă de trafic, s-a încercat folosirea unei arhitecturi paralele asupra aplicațiilor creând arhitecturi orientate pe servicii, iar în momentul în care aceste servicii sunt concentrate pe un număr redus de funcționalități și pot fi lansate independent și să își păstreze funcționalitate putem vorbi de o arhitectură bazată pe microservicii.

Însă schimbând arhitectura sau practiciile din modul de desfășurare al livrării produsului nu este suficient pentru a asigura performanța, întrucât ambele dintre ele aduc dezavantaje ce trebuie tratate iar uneori acestea sunt mai mari decât avantajele pe care schimbarea le-ar aduce, însă implementarea corectă în locurile ce ar beneficia de o astfel de abordare poate să aducă performanțe ce nu ar fi posibile cu procesele vechi.

Afinitatea mea pentru microservicii provine de la lipsa de cunoștințelor pentru a putea alege dintre mai multe limbaje de programare, framework-uri sau platforme. De exemplu, ce avantaje mi-ar face să aleg ca pentru server-ul meu să folosesc o aplicație în Node.JS sau una în Go? Chiar dacă putem să cunoaștem niște lucruri informative la început, cei mai buni indicatori sunt monitorizările proprii în producție. De asemenea pot fi cazuri în care îmbunătățirile aduse unei platforme pe parcurs o fac mai bună față de ce era inițial (dezvoltarea TypeScript poate să influențeze alegerea inițială a unei platforme SpringBoot ce este type-safe). Astfel, pe o arhitectură bazată pe microservicii putem să folosim limbajul potrivit pentru serviciul căruia îi aduce cele mai multe beneficii. Acest lucru se extinde și pentru baze de date. Poate în timpul evoluției serviciului apar tehnologii noi, însă având un singur lucru central ar face migrarea mult mai grea, de asemenea uneori ar trebui să facem compromisuri pe care alegerea unui alt tip de baze de date ar face să o dispară.

Dezvoltarea tematicilor DevOps pornește de la ușurința cu care ne putem crea o bază de date folosind containere, întrucât instalarea unei baze de date pe calculatorul

personal în timp ce lucram la proiectele de facultate a făcut să am experiențe în care la finalul fiecărui an îmi reinstalam sistemul de operare doar pentru că o dată ce instalam o bază de date crea suficiente servicii, foldere care după o dezinstalare încă rămâneau. Folosind Docker, pot crea o bază de date cu o singură linie de cod, care în același timp să fie preconfigurată și să o pot împărtăși cu colegii de proiect ca să nu pierdem timp în configurarea mediului de lucru.

Lucrarea mea va urma o structură în care prezint noțiunile teoretice la început iar la final încerc să aplic aspectele prezentate în crearea unei aplicații. Inițial pornesc cu prezentarea microserviciilor continuând cu diferitele moduri în care acestea pot fi lansate, imediat după cu dezvoltarea noțiunilor de DevOps ce au ca scop accelerarea dezvoltării, iar la final o prezentare generală a modului în care îmi construiesc aplicația.

Capitolul 2

Preliminarii

Capitolul 3

Microservicii

3.1 Concepte generale

3.2 Design

3.3 Tehnici de implementare

3.4 Build

3.5 Testare

3.6 Deployment

3.7 Monitorizare

3.8 Securizare

3.9 Evolutie

Capitolul 4

Orchestrare

4.1 Istoric deployment

4.2 Tipuri de deployment

4.2.1 Baremetal

4.2.2 Mașină virtuală

4.2.3 Container

4.2.4 Function as a Service

4.2.5 Platform as a Service

4.2.6 Container as a Service

4.3 Docker

4.4 Docker Swarm

4.5 Kubernetes

4.6 Red Hat OpenShift

4.7 HashiCorp Nomad

4.8 Infrastructure as a Service

Capitolul 5

Îmbunătățirea metodelor de dezvoltare software

5.1 Schimbari in livrarea produselor

5.2 DevOps și aspecte ale comunitații

5.3 Ce înseamnă să aplici DevOps

5.4 Unelte

Capitolul 6

Aplicație

Capitolul 7

Concluzii