Зборник радова Факултета техничких наука, Нови Сад

УДК: 004.9
ДОИ: https://doi.org/10.24867/28BE24Petrovic

РАЗВОЈ БЕКЕНД ПОДСИСТЕМА БЕЗ СЕРВЕРА ЗА ВЕБ 3.0 ПЛАТФОРМУ ЗА УПРАВЉАЊЕ КУРСЕВИМА

DEVELOPMENT OF A SERVERLESS BACKEND FOR A WEB 3.0 PLATFORM FOR COURSE MANAGEMENT

Марија Петровић, *Факултет техничких наука, Нови Сад*

Област – ЕЛЕКТРОТЕХНИКА И РАЧУНАРСТВО

Кратак садржај – *Рад се бави развојем бекенд подсистема без сервера (енгл. serverless) апликације за управљање курсевима применом рачунарства у облаку. Поред три главна модела услуга (IaaS, PaaS, SaaS) обрађени су и све популарнији концепти serverless и FaaS услуга. Задатак рада је да представи једно од могућих архитектуралних решења које смањује недостатке традиционалних система користећи предности облака и интеграцију са блокчејн компонентом. Детаљно је описан начин имплементације и како се ресурси на Амазоновом облаку (Amazon Web Services) могу креирати помоћу IaC (Infrastructure as Code) концепта.*

Кључне речи: *serverless, платформа, облак, AWS, Ethereum*

Abstract – *This paper focuses on the development of a serverless backend subsystem for a cloud-based application managing courses. In addition to the three main service models (IaaS, PaaS, SaaS), it explores the increasingly popular concepts of serverless and FaaS services. The task of the thesis is to present one of the possible architectural solutions that addresses the drawbacks of traditional systems by leveraging the advantages of the cloud and integration with the blockchain component. The implementation method is detailed, along with how resources on Amazon Web Service (AWS) can be created using the Infrastructure as Code (IaC) concept.*

Keywords: *serverless, platform, cloud, AWS, Ethereum*

1. УВОД

Традиционалне компаније су се ослањале на локалне сервере и инфраструктуру унутар својих простора за складиштење података и обраду информација, што је довело до све веће потребе за одржавањем сервера који опслужују захтеве.
Такође, појавили су се чести захтеви за велика улагања у набавку хардвера, проширивање меморијског капацитета, али и особља задуженог за управљање овим системима.

Све претходно наведено створило је добар темељ за развој рачунарства у облаку (енгл. *Cloud Computing*), који је донео револуцију омогућавајући компанијама да складиште податке и користе ресурсе путем удаљених центара (енгл. *data centers*) који се налазе ван њихове инфраструктуре. [1].

Следећи корак ка оптимизацији употребе ресурса јесте појава архитектуре без сервера (енгл. *serverless*) где је провајдер задужен за алокацију и бригу о ресурсима.

Међутим, иако су предложена бројна решења и стандарди, још увек су се јављали проблеми централизоване архитектуре. Као решење за то, настају децентрализовани системи и блокчејн технологија (енгл. *Blockchain*) као једна њихова имплементација P2P (*peer-to-peer*) мреже [2].

Овај рад се бави демонстрацијом имплементације serverless бекенд (енгл. *backend*) решења чија архитектура обухвата и блокчејн као компоненту.

2. ТЕОРИЈСКЕ ОСНОВЕ

2.1. Рачунарство у облаку

Дистрибуирани системи имплементирани у стварном свету су често комплексни делови софтвера чије су компоненте размештене на више различитих, физички одвојених машина [3].
Рачунарство у облаку је један од најзначајнијих примера оваквих система и односи се на механизме који корисницима обезбеђују услуге и ресурсе преко Интернета [4].
Услуге облака нуде разни провајдери, од којих су свакако најпопуларнији Amazon Web Services (AWS), Microsoft Azure, и Google Cloud Platform (GCP).

2.2. Основе блокчејн технологије

Блокчејн технологија представља дистрибуирани систем који омогућава сигурну размену података путем криптографски повезаних блокова. Када се формира, блок се повезује са претходним стварајући непрекидан ланац (енгл. *chain*) који се не може променити без сагласности свих учесника у мрежи [5]. Свака трансакција (блок), обезбеђена је криптографским методама и валидирана од стране сваког овлашћеног члана мреже, коришћењем консензус алгоритама. Најпознатији алгоритми су: доказ посла (енгл. *Proof of Work - PoW*), доказ улога

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији ментор је био др Душан Гајић, ванр. проф.

(енгл. *Proof of Stake - PoS*), доказ протеклог времена (енгл. *Proof of Elapsed Time - PoET*) …

2.3. Ethereum блокчејн

Ethereum је децентрализована, јавна блокчејн платформа заснована на *peer-to-peer* мрежи, која омогућава безбедно извршавање и верификацију паметних уговора. Њен креатор је Виталик Бутерин, објавивши 2013. године Ethereum whitepaper који поставља основу за развој децентрализованих платформи и апликација које би подржавале паметне уговоре. Стога, у Ethereum екосистем се уводи рачунарски механизам назван Ethereum виртуелна машина који је централни део Ethereum блокчејна. Она представља извршно окружење за паметне уговоре и управља стањем овог блокчејна.

Сматра се да је Ethereum направио револуцију и када је у питању развој Интернета, уводећи Веб 3.0 генерацију, и тиме подржао децентрализоване апликације (DApps), децентрализоване финансије (DeFi) и децентрализоване размене (DEXs) [6].

Крипто валута која се користи у Ethereum мрежи се назива етер (енгл. *ether – ETH*). Етер има и неколико фракција, тј. мањих јединица, а најважнији су Wei и Gwei.

Ethereum је дуго користио доказ посла (енгл. *Proof of Work*) као консензус алгоритам, међутим од 2022. године прелази на доказ улога (енгл. *Proof of Stake*), који је енергетски ефикаснији и сигурнији механизам.

3. РАЧУНАРСТВО У ОБЛАКУ И SERVERLESS АРХИТЕКТУРА

3.1. Историја рачунарства у облаку

Рачунарство у облаку креће са својим значајнијим развитком шездесетих година прошлог века када су постављени фундаментални концепти – испоручивање сервиса тако да га други могу користити, могућност дељења компјутерских ресурса између више особа и приступ сервисима (ресурсима) путем мреже [7].
Међутим, рани почеци имплементације облака озбиљније крећу тек 1970. године када је компанија IBM лансирала своју прву виртуалну машину – VM/370. Тада је настала виртуализација као фундаментална технологија која стоји иза сваког облака.
Деценију касније појавом виртуалних приватних мрежа (енгл. *Virtual Private Network - VPN*) постаје могуће повезивање рачунара у просторијама како би људи могли да приступе дељеним подацима. Све већом производњом компјутерских ресурса цене су почеле да падају, а услуге да расту, тако да се створио темељ да почетком 21. века изађе и први јавни провајдер облака – Amazon Web Services (AWS).

3.2. Сервисни модели облака

Како је рачунарство у облаку временом добијало све већу популарност издвојило се неколико различитих модела који задовољавају специфичне потребе корисника, а то су инфраструктура као сервис (енгл. *Infrastructure as a Service – IaaS*), платформа као сервис (енгл. *Platform as a Service – PaaS*) и софтвер као сервис (енгл. *Software as a Service – SaaS*).

3.2. Serverless рачунарство и архитектуре

Коришћењем услуга провајдера рачунарства у облаку смањује се проблем бриге о хардверу али и даље се захтева одређени ниво управљања осталим софтверима на серверу, скалирањем и ценом [8].

Serverless рачунарство представља један модел рачунарства у облаку који функционише тако што провајдер алоцира ресурсе на захтев, водећи бригу о серверима уместо корисника. Сви ресурси се плаћају онолико колико се користе (енгл. *pay-as-you-go*).

3.3. Функција као сервис (Function as a Service – FaaS)

Иако се често користе у истом контексту, функција као сервис је у ствари један вид имплементације serverless модела који омогућава покретање функција написаних у различитим програмским језицима у облаку. Графички приказ уобичајеног тока догађаја приликом рада са serverless функцијама је дат на слици 1.

Слика 1. Приказ како serverless функција ради [9].

Пошто FaaS представља један подскуп serverless модела од њега наслеђује предности и мане.

3.4. Пример облака: Amazon Web Services (AWS)

AWS представља обимну и једну од најзаступљенијих платформи за рачунарство у облаку, насталу под окриљем компаније Амазон, и обухвата широк спектар услуга који подржавају раније поменуте моделе облака – IaaS, PaaS, SaaS, понуђене у форми различитих сервиса [10]. Такође, један је од првих провајдера који су увели *pay-as-you-go* модел коришћења облака.

AWS данас има центре података распрострањене на преко 87 зона доступности (енгл. *Availability Zone – AZ*) у регионима широм света који AWS чине скалабилном и високо доступном платформом.

У свом екосистему, AWS нуди велики број сервиса за различите примене и архитектуре, а неки од њих су: Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3), Amazon Relational Database Service (RDS), Amazon Lambda, Amazon API Gateway …

3.5. AWS Cloud Development Kit (AWS CDK)

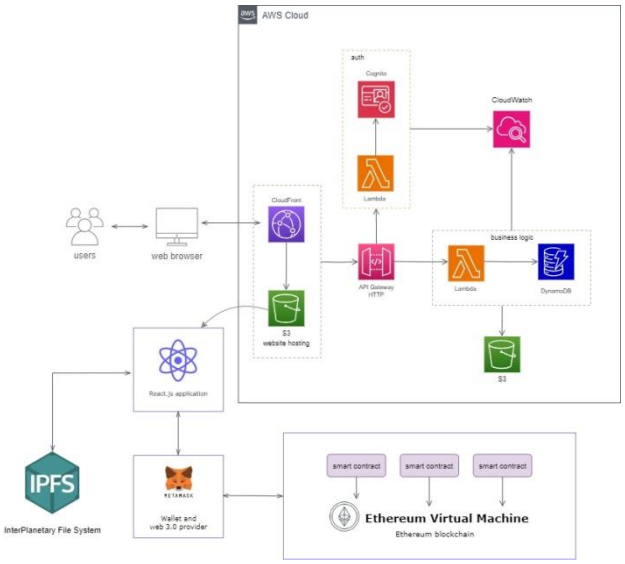
Када се поставља инфраструктура неког система важно је организовати компоненте на добар начин

како би се ефикасније и агилније могло управљати њоме. Како су временом комплексност и размере инфраструктуре расле било је јасно да традиционални приступ са својим недостацима више није погодно решење. То је био моменат када почиње да се развија концепт под називом инфраструктура као код (енгл. *Infrastructure as Code – IaC*) који омогућава управљање читавом инфраструктуром помоћу конфигурационих фајлова написаних у коду [11].

AWS Cloud Development Kit представља радни оквир за дефинисање и обезбеђивање инфраструктурних AWS ресурса користећи познате програмске језике.

4. АРХИТЕКТУРА СИСТЕМА

Платформа је развијена као децентрализована веб 3.0 апликација, са одвојеним клијентским и серверским делом који остварују међусобну комуникацију путем HTTPS протокола. И клијентска и серверска апликација су постављене на AWS облак – S3 сервис је употребљен за хостовање статичког веб сајта, док је бекенд развијен као serverless апликација (слика 3).



Слика 3. Дијаграм архитектуре решења.

Клијентска страна је имплементирана коришћењем React.js библиотеке у оквиру које је употребљен web3.js скуп библиотека захваљујући коме је омогућена интеграција са дигиталним новчаницима и веб 3.0 провајдерима (у овом случају MetaMask). Апликација комуницира и са IPFS (енгл. *InterPlanetary File System*) мрежом, ради складиштења датотека које је корисник унео, као и са чворовима блокчејн мреже, како би позивала функције паметног уговора за креирање и добављање корисникових незамењивих токена.

Бизнис логика серверског дела је представља Lambda функцијама које комуницирају са NoSQL базом података, DynamoDB. Када се захтев упути серверу он прво стиже на API Gateway који ће на основу детаља из захтева окинути одговарајућу Lambda функцију. Ауторизација и аутентификација корисника је имплементирана коришћењем Cognito сервиса. За мониторинг Lambda функција у решење је интегрисан и AWS CloudWatch сервис чији је задатак да прати и прикупља метрике коришћених ресурса.

Компонента која чини апликацију децентрализованом јесте Ethereum блокчејн мрежа над којом се извршавају паметни уговори.

5. ИМПЛЕМЕНТАЦИЈА РЕШЕЊА

Обе апликације (клијентска и серверска) су смештене у AWS окружењу, а инфраструктурни ресурси су подигнути уз помоћ AWS Cloud Development Kit радног оквира са којим је веза остварена кроз програмски језик Go. Сви ресурси се налазе у eu-central-1 региону.

5.1. Клијентска апликација

Клијентска страна представља *single-page* апликацију (енгл. SPA – *Single page application*) и доступна је на Интернету захваљујући интеграцији два сервиса – Amazon S3 и Amazon CloudFront. Комбинација две поменуте платформе је изузетно честа и може се рећи да представља стандард за испоруку фронтенд апликација.

5.2. Серверска апликација – serverless решење

Lambda функције представљају језгро инфраструктуре серверске стране јер са собом носе код специфичан за одређени део пословне логике. Пројекат је организован тако да је свака функционалност представљена са по једном Lambda функцијом, у засебном .go фајлу. На листингу 1 је дата генеричка структура коју поштује сваки од фајлова из lambda пакета.

```
func Handler(ctx context.Context, req
events.APIGatewayV2HTTPRequest)
(events.APIGatewayV2HTTPResponse, error) {
    // business logic
}

func main() {
    lambda.Start(Handler)
}
```

Листинг 1. Структура фајла једне Lambda функције.

Одговарајућа Lambda ће се окинути када са клијентске стране стигне захтев на API Gateway намењен за њу. Након што функција заврши процесирање вратиће на API Gateway одговор који ће бити прослеђен крајњем кориснику. Како би обавила своју функционалност, Lambda функција позива методе из одговарајућих сервиса који комуницирају са слојем намењеним за приступ бази података.

Да би се ресурси сместили на AWS облак то десило потребно је покренути следећу команду у терминалу:

```
cdk deploy
```

Да би ова команда имала ефекта и да би уопште могла да се изврши, мора да постоји конфигурациони фајл `cdk.json`. Пример фајла је дат на листингу 2.

```
{
  "app": "go mod download && go run
backend.go",
  "watch": {
    "include": [
      "**"
    ],
    "exclude": [
      "cdk*.json",
      "go.mod",
      "go.sum"
    ]
  },
  "context": {
    "@aws-cdk/aws-
lambda:recognizeLayerVersion": true,
    "@aws-cdk/core:checkSecretUsage": true

    // the rest of the code is omitted
  }
}
```

Листинг 2. Пример `cdk.json` *датотеке.*

Датотека се генерише при иницијализацији CDK пројекта (уз команду **`cdk init`**) и садржи наредбе неопходне да се инфраструктура покрене и испоручи (енгл. *deploy*) у AWS окружењу.

6. ЗАКЉУЧАК

Рад се бавио применом концепата рачунарства у облаку и serverless технологије као део њега, како би се имплементирао систем за управљање курсевима. Применом једне од популарних платформи облака (AWS) као и serverless архитектуре главне предности система јесу ефикасност трошкова и оптимизација ресурса. Основни недостатак система из блокчејн перспективе јесте присуство традиционалне серверске апликације која има посредничку улогу између клијентског интерфејса и паметних уговора. Са стране облака може да се истакне безбедност као главна брига, јер су подаци поверени провајдерима услуга, те снажне безбедоносне мере морају постати императив. Појава праксе развоја у облаку је постала данас битан фактор у модерном развоју софтвера. Пројекат описан у раду је само један од примера трансформације традиционалних архитектуралних решења под утицајем рачунарства у облаку. Ово поље рачунарства је у константном процесу еволуције, а нове технологије попут рачунарства на рубу (енгл. *edge computing*), serverless архитектуре, блокчејна и услуга покретаних вештачком интелигенцијом су у стању да преобликују хоризонт могућности.

7. ЛИТЕРАТУРА

[1] S. J. Bigelow, "TechTarget," 15 November 2022. [На мрежи]. Доступно: https://www.techtarget.com/whatis/feature/The-history-of-cloud-computing-explained. [Последњи приступ децембар 2023].

[2] "modex.tech," 6 December 2021. [На мрежи]. Доступно: https://www.modex.tech/blog/centralized-vs-decentralized-vs-distributed-systems. [Последњи приступ децембар 2023].

[3] M. v. Steen, Distributed Systems (4th edition).

[4] Montazerolghaem, Ahmadreza, Yaghmaee, M. Hossein, Leon-Garcia and Alberto, Green Cloud Multimedia Networking: NFV/SDN Based Energy-Efficient Resource Allocation, 2020.

[5] A. S. Ravikiran, "simplelearn.com," October 2023. [На мрежи]. Доступно: https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-technology. [Последњи приступ децембар 2023].

[6] "Evolution of Blockchain Technology," 2 November 2022. [На мрежи]. Доступно: https://www.nationthailand.com/business/corporate/40021641. [Последњи приступ децембар 2023].

[7] B. Varghese, "History of the cloud," 19 March 2019. [На мрежи]. Доступно: https://www.bcs.org/articles-opinion-and-research/history-of-the-cloud/. [Последњи приступ децембар 2023].

[8] "What is serverless computing?," [На мрежи]. Доступно: https://www.cloudflare.com/learning/serverless/what-is-serverless/. [Последњи приступ јануар 2024].

[9] "Serverless Architecture Overview," [На мрежи]. Доступно: https://www.datadoghq.com/knowledge-center/serverless-architecture/. [Последњи приступ јануар 2024].

[10] N. Barney, "Amazon Web Services," [На мрежи]. Доступно: https://www.techtarget.com/searchaws/definition/Amazon-Web-Services. [Последњи приступ јануар 2024].

[11] "What Is Infrastructure as Code?," [На мрежи]. Доступно: https://stackify.com/what-is-infrastructure-as-code-how-it-works-best-practices-tutorials/. [Последњи приступ јануар 2024].

Кратка биографија:

Марија Петровић је рођена 9. априла 1999. године у Новом Саду. Факултет техничких наука у Новом Саду, студијски програм Рачунарство и аутоматика уписала је 2018. године. Након завршених основних студија, 2022. године, уписала је мастер академске студије из исте области на смеру Рачунарство високих перформанси.
контакт:
petrovic.ma9@gmail.com