

**ILUSTRACIJA PRIMENE AZURE PLATFORME I BICEP JEZIKA NA PRIMERU
APLIKACIJE ZA PLANIRANJE VREMENA I KREIRANJE IZVEŠTAJA****ILLUSTRATION OF THE USE OF THE AZURE PLATFORM AND BICEP LANGUAGE
ON A TIME PLANNING AND CREATION APPLICATION**

Jovica Josipović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – U ovom radu analiziraju se ključni koncepti i modeli računarstva u oblaku, uključujući prednosti i izazove njegove primene. Poseban fokus je na Infrastrukturi kao kodu (IaC), sa detaljnim pregledom korišćenog Bicep jezika. Opisane su važne Azure usluge za razvoj i upravljanje aplikacijama. Implementacija rešenja obuhvata planiranje, kreiranje izveštaja, i konfiguraciju YAML pipeline-a za automatizaciju procesa. Na kraju, rad prikazuje faze prevođenja, testiranja i objavljivanja infrastrukture i aplikacija na Azure platformi.

Ključne reči: Računarstvo u oblaku, Infrastruktura kao kod, Microsoft, Azure, Bicep, Application Service, ARM šabloni

Abstract – This paper analyzes key concepts and models of cloud computing, including the advantages and challenges of its implementation. It places special focus on Infrastructure as Code (IaC) and provides a detailed overview of the Bicep language. Important Azure services for application development and management are also described. The solution implementation includes planning, report creation, and the configuration of YAML pipelines for process automation. Finally, the paper outlines the phases of building, testing, and deploying infrastructure and applications on the Azure platform.

Keywords: Cloud computing, Infrastructure as Code, Microsoft, Azure, Bicep, Application Service, ARM templates

1. UVOD

Razvoj i upotreba računarstva u oblaku [1] značajno su promenili pristup kompanija infrastrukturi i održavanju aplikacija. Ovaj model pruža fleksibilnost, skalabilnost i ekonomičnost u upravljanju resursima. U radu se fokusiramo na implementaciju infrastrukture kao koda (IaC) kao ključne komponente automatizacije u oblaku. Posebna pažnja posvećuje se Microsoft Azure platformi i upotrebi Bicep jezika i ARM šablona. Cilj rada je demonstracija primene ovih tehnologija u aplikacijama za planiranje vremena i upravljanje zadacima.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Gordana Milosavljević, red. prof.

2. RAČUNARSTVO U OBLAKU

Računarska tehnologija u oblaku obuhvata skup računarskih usluga i resursa koji se isporučuju preko interneta. Ova tehnologija omogućava organizacijama i pojedincima da pristupe i koriste računarske resurse i usluge putem interneta. Umesto da aplikacije i podaci budu smešteni na lokalnim serverima ili ličnim računarima, računarstvo u oblaku korisnicima omogućava pristup računarskoj snazi, skladištenju i softverskim aplikacijama preko udaljenih centara podataka. Ovaj model računarstva funkcioniše na osnovu plaćanja prema stvarnoj potrošnji, gde korisnici plaćaju samo za resurse koje zaista koriste, pružajući ekonomična i skalabilna rešenja.

2.1. Modeli implementacije

Računarstvo u oblaku nudi različite modele implementacije kako bi se odgovorilo na različite potrebe poslovnih organizacija. Osnovni modeli implementacije su:

- Privatni oblak – Privatni oblak je posvećen jednoj organizaciji.
- Javni oblak – Javni oblak pruža usluge većem broju organizacija.
- Hibridni oblak – Hibridni oblak kombinuje elemente privatnih i javnih oblaka.

2.2. Modeli usluga

Računarstvo u oblaku podržava različite modele usluga [2]:

- Infrastruktura kao servis (IaaS - engl. *Infrastructure as a Service*) – pruža virtualizovane računarske resurse, kao što su virtuelne mašine, skladište i umrežavanje, omogućavajući korisnicima kontrolu nad operativnim sistemom i aplikacijama.
- Platforma kao servis – nudi platformu sa ugrađenim alatima za razvoj i usluge, omogućavajući razvojnim inženjerima da se fokusiraju na pisanje i implementaciju aplikacija bez upravljanja infrastrukturom.
- Softver kao servis (SaaS - engl. *Software as a Service*) – aplikacije su smeštene u oblaku i dostupne preko interneta na osnovu pretplate, pružajući praktičnost i eliminišući potrebu za instalacijom i održavanjem.

Glavne prednosti upotrebe računarstva u oblaku su:

- Ušteda troškova
- Skaliranje i fleksibilnost
- Pristupačnost i mobilnost
- Pouzdanost i visoka dostupnost
- Bezbednost i usklađenost.

3. INFRASTRUKTURA KAO KOD

Infrastruktura kao kod je postala revolucionarni koncept koji preoblikuje način na koji organizacije upravljaju svojom IT infrastrukturom [3]. Ovaj princip predstavlja fundamentalnu promenu u načinu razmišljanja i samom razvoju digitalne strukture jedne organizacije. Infrastruktura kao kod je pristup koji tretira elemente infrastrukture, kao što su serveri, mreže i baze podataka, kao da su softver. Ovaj pristup omogućava preduzećima da tretiraju svoju infrastrukturu kao softver, omogućavajući automatizovano i efikasno upravljanje, što na kraju dovodi do povećane produktivnosti i smanjenja operativnih troškova.

3.1. Ključni principi infrastrukture kao koda

Osnovni principi infrastrukture kao koda su:

- Deklarativna konfiguracija – IaC se oslanja na deklarativni kod koji specificira željeno stanje infrastrukture, umesto da propisuje niz koraka za ručnu konfiguraciju.
- Automatizacija – Automatizacija je srž IaC-a. Automatizuje nabavku, konfiguraciju i skaliranje resursa infrastrukture, smanjujući ljudske greške i omogućavajući brzu i pouzdanu implementaciju.
- Kontrola verzija – IaC kodom se upravlja u sistemima za kontrolu verzija kao što je Git, olakšavajući saradnju, praćenje promena i obezbeđivanje istorije zapisa konfiguracija.
- Nepromenljiva infrastruktura – IaC promovise stvaranje novih instanci infrastrukture umesto modifikacije postojećih.

Glavne prednosti upotrebe infrastrukture kao koda su:

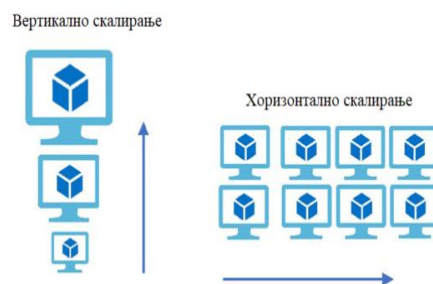
- Agilnost i brzina
- Doslednost i pouzdanost
- Optimizacija troškova
- Obnova nakon katastrofe.

3.2. Microsoft Azure servis

Azure je Microsoft-ova platforma za računarstvo u oblaku i sveobuhvatni paket usluga zasnovanih na oblaku. Azure pruža moćnu podlogu za preduzeća svih veličina, od startup-a do multinacionalnih kompanija. Jedna od izuzetnih karakteristika Azure je njegova skalabilnost. Organizacije mogu iskoristiti infrastrukturu kao uslugu i platformu kao uslugu kako bi se lako skalirale prema gore (vertikalno) ili prema dole (horizontalno) u skladu sa promenama svojih potreba. Na slici 1 dat je prikaz vertikalnog i horizontalnog skaliranja.

3.3. Tradicionalni jezici infrastrukture kao koda

Infrastruktura je definisana i upravljana pomoću koda. Postoji nekoliko tradicionalnih i popularnih jezika i alata koji se koriste za implementaciju infrastrukture kao koda [4].



Slika 1. Vertikalno i horizontalno skaliranje

Najpopularniji predstavnici su:

- Terraform
- Google Cloud Deployment Manager
- AWS CloudFormation
- Azure Resource Manager (ARM) Templates.

3.4. Bicep jezik

Bicep predstavlja Microsoft-ov jezik koji je kreiran za definisanje i implementaciju Azure resursa, pružajući jednostavniju i efikasniju sintaksu za ARM šablone. Ovaj jezik uvodi novu ekstenziju fajla, ".bicep", čime se udaljava od tradicionalnih ".json" fajlova koji se koriste za ARM šablone [5]. Putem procesa kompilacije, Bicep fajlovi se pretvaraju u ekvivalentne ARM JSON šablone, koji su potpuno usklađeni sa Azure infrastrukturom. Ovaj pojednostavljeni pristup ne samo da olakšava pisanje koda, već i poboljšava integraciju Bicep-a u Azure ekosistem, doprinoseći bržem i preciznijem upravljanju resursima. Uporedimo generalne osobine Bicep jezika i ARM šablona [6].

- Čitljivost – Bicep je poznat po poboljšanoj čitljivosti i sažetosti. Koristi deklarativnu i intuitivniju sintaksu u poređenju sa ARM šablonima, što ga čini lakšim za razumevanje i održavanje. ARM šabloni koriste sintaksu baziranu na JSON-u, koja može biti opsežna i složena. Razumevanje i modifikacija velikih ARM šablona može biti izazovno zbog obimne upotrebe JSON-a.
- Sažetost – Bicep omogućava da se definiše infrastruktura korišćenjem jezika višeg nivoa na sažetiji način. Smanjuje količinu uobičajenog koda u poređenju sa ARM šablonima. ARM šabloni, kao JSON, mogu biti opsežniji. Ova opširnost može učiniti šablone dužim i težim za čitanje, posebno kod složenih implementacija.
- Lakša upotreba – Bicep se smatra jednostavnim za upotrebu (eng. user friendly) i orijentisanim ka programerima. Njegova sintaksa je dizajnirana da bude intuitivnija, a pruža i funkcionalnosti kao što su upotreba tipova, što znatno olakšava rad. Rad direktno sa ARM šablonima može zahtevati veće razumevanje tehnologija, posebno za one manje upućene u JSON sintaksu. Debugovanje (eng. debugging) i rešavanje problema takođe mogu biti izazovni.
- Podrška za alate – Bicep ima svoj set alata i podržava funkcionalnosti kao što su pametno dovršavanje koda, isticanje sintakse i provera

grešaka. To poboljšava iskustvo razvoja. Iako postoje alati i proširenja za rad sa ARM šablonima, oni možda ne pružaju isti nivo olakšanja rada i praktičnosti kao alati za Bicep.

4. KLJUČNE AZURE USLUGE ZA RAZVOJ I UPRAVLJANJE APLIKACIJAMA

Azure nudi sveobuhvatan skup usluga koje su ključne za razvoj i upravljanje aplikacijama. Ove usluge omogućavaju da se lako grade, postavljaju i održavaju robusne, skalabilne aplikacije. Usluge koje su korišćene prilikom izrade infrastrukture su:

- Azure aplikativni servis (*engl. Application Service*) – Platforma koju nudi Microsoft Azure, omogućavajući brzo i efikasno razvijanje, implementaciju i skaliranje veb aplikacija i API-ja. Ona apstrahuje osnovnu infrastrukturu, omogućavajući inženjerima da se fokusiraju na svoj kod bez brige o upravljanju serverima ili mrežom.
- Azure skladište (*engl. Azure Storage*) – Platforma za brzo i lako skladištenje različitih vrsta podataka. Azure skladište obezbeđuje različita rešenja za skladištenje, dizajnirana da zadovolje raznovrsne potrebe poslovanja u digitalnom dobu.
- Azure trezor (*engl. Azure Key Vault*) – Omogućava bezbedno skladištenje i upravljanje osetljivim informacijama kao što su ključevi, tajne, sertifikati i web povezivanja.
- Azure monitor (*engl. Azure Application Insights*) – Obezbeđuje kompletnu uslugu za praćenje i upravljanje performansama aplikacija.

5. IMPLEMENTACIJA APLIKACIJE ZA PLANIRANJE VREMENA I KREIRANJE IZVEŠTAJA

Aplikacija “Planiranje vremena i kreiranje izveštaja” napravljena je radi pojednostavljivanja raspoređivanja zadataka, praćenje utrošenog vremena i generisanja izveštaja na efikasan i lako upotrebljiv način. Osnovne funkcionalnosti aplikacije su:

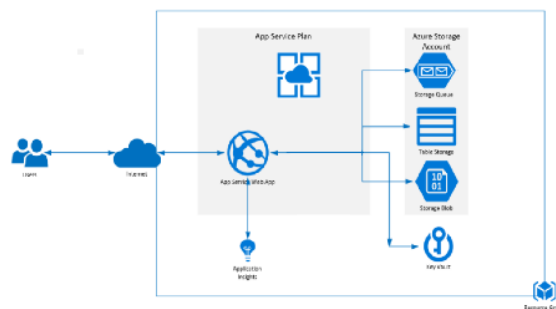
- Planiranje vremena, koje omogućava korisnicima da rasporede zadatke u određenom vremenskom periodu, vodeći računa o usklađenosti sa radnim vremenom i rokovima.
- Kreiranje detaljnih izveštaja u Excel formatu koji pokazuju koliko je vremena utrošeno na svaki zadatak, uz prikaz dnevnih detalja i sažetaka za pregled.

Aplikacija je razvijena uz pomoć C# i EPPlus biblioteke za generisanje Excel fajlova, dok se Azure Blob Storage koristi za bezbedno čuvanje i pristup generisanim tabelama. Podaci su organizovani u sledećoj strukturi:

- *Kontejneri* – Svaki korisnik ima svoj kontejner za čuvanje izveštaja.
- *Fajlovi* – Izveštaji se čuvaju u formatu Timesheet-GGGG-MM-DD.xlsx, gde svaki fajl sadrži podatke o zadacima za određeni period. Moguće je izvršiti modifikaciju imena fajla.

Logika aplikacije osigurava efikasnu dodelu zadataka i praćenje vremena u skladu sa poslovnim procesima. Na

slici 2 je prikazana arhitektura aplikacije za planiranje vremena i kreiranje izveštaja.



Slika 2. Arhitektura aplikacije za planiranje vremena i kreiranje izveštaja

5.1 YAML pipelines

YAML je format za serijalizaciju podataka koji je lak za čitanje i koristi se u Azure Pipelines za definisanje radnih tokova prevođenja i objavljivanja. YAML radni tokovi nude nekoliko prednosti:

- Kontrola verzija – Konfiguracije YAML radnih tokova mogu biti sačuvane uz izvorni kod, pružajući istoriju verzija i praćenje promena.
- Fleksibilnost – YAML omogućava složene definicije radnih tokova sa uslovnom logikom, šablonima i ponovo upotrebljivim komponentama.
- Održivost – Deklarativna priroda YAML-a olakšava razumevanje i održavanje konfiguracija radnih tokova u poređenju sa klasičnim pristupom zasnovanim na korisničkom interfejsu.

5.2 Faza infrastrukture

Kreiranje infrastrukture je ključna komponenta procesa raspoređivanja koja uključuje podešavanje i konfigurisanje neophodnih resursa u oblaku za aplikaciju. Na slici 3 je prikazana faza infrastrukture.

✓	Create Infrastructure	4m 24s
⌚	Initialize job	1s
✓	Checkout Master@main to s	7s
✓	Create Resource Group	1m 9s
✓	Create Storage Account	1m 13s
✓	Create Key Vault	54s
✓	Create App Service	56s
✓	Post-job: Checkout Master@mai...	<1s
⌚	Finalize Job	<1s

Slika 3. Faza infrastrukture

5.3 Faza prevođenja i testiranja

Faze prevođenja i testiranja su ključne komponente CI/CD pipeline-a. One osiguravaju da aplikacija bude pravilno kompajlirana i testirana, pružajući kontinuirane povratne informacije o kvalitetu koda i funkcionalnosti. Na slici 4 je prikazana faza prevođenja i testiranja.

▼	Build and Test .NET Application	1m 49s
✓	Initialize job	1s
✓	Checkout Master@main to s	4s
✓	Restore NuGet packages	1m 10s
✓	Build the project	15s
✓	Run unit tests	1s
✓	Run unit tests	8s
✓	PublishTestResults	6s
✓	Post-job: Checkout Master@mai...	<1s
✓	Finalize Job	<1s

Slika 4. Faza prevođenja i testiranja

5.4 Faza objavljivanja

Faza objavljivanja je proces postavljanja aplikacije u produkciono okruženje, omogućavajući korisnicima pristup novim funkcijama i poboljšanjima. Na slici 5 je prikazana faza objavljivanja.

▼	Deploy .NET Application	1m 45s
✓	Initialize job	3s
✓	Checkout Master@main to s	4s
✓	Publish the project	1m 0s
✓	List files in publish directory	2s
✓	Zip files in container	13s
✓	Deploy to App Service	20s
✓	Post-job: Checkout Master@mai...	<1s
✓	Finalize Job	<1s

Slika 5. Faza objavljivanja

6. ZAKLJUČAK

U ovom radu je detaljno, u edukativne svrhe, obrađena Azure platforma, a u svrhu ilustracije njene upotrebe, napravljena je i detaljno prikazana aplikacija za planiranje vremena i kreiranje izveštaja. Ovaj projekat pokazuje kako savremeni alati u oblaku i tehnologije, uz pravilno upravljanje infrastrukturom i razvojnim procesima, doprinose izgradnji sigurnih, stabilnih i skalabilnih aplikacija koje lako odgovaraju na zahteve savremenog poslovanja.

Dalji razvoj aplikacije obuhvata poboljšanje korisničkog interfejsa aplikacije i bolje korisničko iskustvo. Planira se integracija novih Azure alata za bolju obradu podataka i prediktivno planiranje.

7. LITERATURA

- [1] Edge, Fog i Cloud računarstvo, <http://cs1.ftn.kg.ac.rs/edge-cloud/>
- [2] Cloud Stack Consideration: Choosing Between SaaS, PaaS, and IaaS. <https://www.bcsconsultants.com/blog/saas-paas-iaas/>, 2016.
- [3] Microsoft Learn - What is infrastructure as code (IaC)? <https://learn.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code>, 2022.
- [4] Saurabh Dashora - Infrastructure as Code vs Traditional Infrastructure, <https://progressivecoder.com/infrastructure-as-code-vs-traditional-infrastructure/>, 2023.
- [5] Thomas Thornton - Bicep for Azure – The Future of Infrastructure as Code, 2022
- [6] Ritesh Modi - Infrastructure as Code with Azure Bicep, Packt Publishing, 2022.
- [7] Kief Morrison - Infrastructure as Code: Managing Servers in the Cloud 1st Edition, O'Reilly Media, 2016.

Kratka biografija:



Jovica Josipović rođen je 4.9.1997. u Prijedoru, BiH. Završio je osnovne akademske studije 2020. godine na Fakultetu tehničkih nauka u Novom Sadu. Upisao je master studije iste godine, studijski program Računarsvo i automatika - Elektronsko poslovanje.