Зборник радова Факултета техничких наука, Нови Сад

UDK: 4.9
DOI: https://doi.org/10.24867/30BE06Njegic

# *CLOUDCAST* – ПЕРСОНАЛИЗОВАНЕ НОТИФИКАЦИЈЕ ЗА ПРОГНОЗУ ВРЕМЕНА ЗАСНОВАНЕ НА *AWS SERVERLESS* ИНФРАСТРУКТУРИ

# CloudCast – PERSONALIZED WEATHER FORECAST NOTIFICATIONS BASED ON AWS SERVERLESS INFRASTRACTURE

Ђорђе Његић, *Факултет техничких наука, Нови Сад*

**Област – ЕЛЕКТРОТЕХИНЧКО И РАЧУНАРСКО ИНЖЕЊЕРСТВО**

**Кратак садржај –** *У овом раду описан је CloudCast, сервис заснован на претплати који пружа персонализоване временске нотификације путем е-поште или СМС-а. Корисници конфигуришу свој налог на основу чега добијају персонализоване нотификације на дневном нивоу са информацијама о тренутним временским условима и прогнозама. Сервис је у потпуности изграђен на AWS инфраструктури користећи serverless архитектуру.*

**Кључне речи:** *Претплатничке услуге, Serverless архитектура, AWS инфраструктура*

**Abstract –** *This paper describes CloudCast, a subscription-based service that provides personalized weather notifications via email or SMS. Users configure their accounts to receive daily personalized notifications with information about current weather conditions and forecasts. The service is entirely built on AWS infrastructure using a serverless architecture.*

**Keywords:** *Subscription Services, Serverless Architecture, AWS Infrastructure*

## 1. УВОД

Данас постоји велики број сервиса који пружају информације о временским условима. Ови сервиси су углавном доступни путем мобилних и веб апликација, омогућавајући корисницима да брзо и лако добију прогнозу времена. Међутим, већина ових сервиса пружа само основне податке и то на кориснички захтев, што може бити непрактично за оне који су у покрету или имају ограничен приступ интернету.

Оно што разликује сервис *CloudCast* који ће бити представљен у овом раду јесте начин информисања корисника путем претплате и дневних нотификација. Корисници овог сервиса добијају персонализоване нотификације које су прилагођене њиховим потребама у погледу времена када их добијају, садржаја нотификације, као и медијума путем којег се обавештавају (е-маил или СМС).

---

**НАПОМЕНА:**
**Овај рад проистекао је из мастер рада чији ментор је био др Мирослав Зарић, ред. проф.**

## 2. КОРИШЋЕНЕ ТЕХНОЛОГИЈЕ

### 2.1. Terraform

*Terraform* [1] је алат за инфраструктурно кодирање (IaC) који омогућава дефинисање и управљање инфраструктуром кроз декларативне конфигурационе фајлове. Користи се за аутоматизацију креирања и управљања ресурсима у cloud окружењима као што су AWS, Azure, Google Cloud и други. За разликзу од IaC алата које нуде сами cloud-provider-и, Terraform је независан од конкретног окружења и омогућава дефинисање инфраструктуре јединственом синтаксом за све. Такође омогућава верзионисање инфраструктуре, што олакшава праћење промена и враћање на претходне верзије уколико је потребно.

### 2.2. AWS (Amazon Web Services)

*AWS* [2] је водећи глобални понуђач услуга у облаку основан 2006. гоидине. Нуди широк спектар услуга у оквиру више од 200 различитих сервиса. Неке од функционалности које овај сервис нуди су креирање и управљање виртуелним серверима, складиштење и обрада података, базе података, мониторинг, аналитика, машинско учење и друго. AWS омогућава организацијама да брзо и ефикасно развијају, имплементирају и скалирају апликације без потребе за управљањем инфраструктуром.

#### 2.2.1 AWS API Gateway

*AWS API Gateway* [3] је сервис који омогућава креирање, одржавање и заштиту API-ја. Омогућава дефинисање различитих типова API-ја, пружајући могућности аутентификације, ауторизације, ограничавања протока и праћења перформанси, скалирање у складу са количином саобраћаја, кеширање, и друго. Интеграција са AWS Lambda функцијама омогућава *serverless* архитектуру, где се пословна логика најчешће извршава у Lambda функцијама као одговор на захтеве који пристигну на API Gateway. Ово представља уобичајен шаблон у дефинисању *serverless* архитектура.

#### 2.2.2 AWS Cognito

*AWS Cognito* [4] је сервис за управљање идентитетима и аутентификацијом корисника који омогућава једноставну интеграцију са другим AWS сервисима попут API Gateway-а. Путем овог сервиса корисници могу да креирају налоге, пријављују се, ресетују лозинке и управљају својим корисничким профилима.

### 2.2.3 AWS Lambda

*AWS Lambda* [5] је сервис који представља основ serverless архитектуре на AWS-у. Омогућава извршавање кода као одговор на догађаје без потребе за управљањем инстанцама сервера. Lambda функције се могу покренути као одговор на различите догађаје, као што су промене у S3 bucket-има, упити ка API Gateway-у, или поруке у SNS темама. Lambda омогућава аутоматско скалирање и плаћање само за утрошено време извршавања кода, што може значајно смањити трошкове у поређењу са инфраструктуром базираном на серверима. Интеграција са другим AWS сервисима омогућава изградњу комплексних апликација без потребе за управљањем инфраструктуром.

### 2.2.4 AWS DynamoDB

*AWS DynamoDB* [6] представља serverless NoSQL базу података којом у потпуности управља AWS, захваљујући чему гарантује брзе и предвидиве перформансе са аутоматским скалирањем. Такође, гарантује високу доступност и отпорност на грешке, што је кључно за апликације које увек морају бити доступне. Подржава сложене упите и индексирање, што олакшава брзо претраживање и филтрирање података. Представља један од стандардних компоненти serverless шаблона на AWS-у.

### 2.2.5 AWS S3 (Simple Storage Service)

*AWS S3* [7] је сервис за складиштење објеката који пружа скалабилно, трајно и сигурно складиштење података. Омогућава једноставно управљање складиштењем података, са могућностима верзионисања, енкрипције и контроле приступа. Често се користи за складиштење статичких фајлова, као што су слике, видео записи, резервне копије и слично. Интеграција са другим AWS сервисима омогућава аутоматизацију процеса, као што је покретање Lambda функција као одговор на специфициране догађаје у неком S3 bucket-у.

### 2.2.6 AWS SES (Simple Email Service)

*AWS SES* [8] је сервис који омогућава слање велики број различитих типова мејлова. Омогућава једноставну интеграцију са другим AWS сервисима, као и аутоматизацију процеса слања мејлова, могућност дефинисања шаблона за мејлове и друго. SES се уобичајено користи за различите верификације, слање нотификација, потврда, *newsletter*-а и других врста email комуникације.

### 2.2.7 AWS SNS (Simple Notification Service)

*AWS SNS* [9] је сервис за слање порука и нотификација у реалном времену. Омогућава слање порука путем различитих медијума, као што су SMS, email и HTTP/HTTPS комуникација, што пружа флексибилност у начину испоруке обавештења.

Често се користи за слање нотификација о догађајима, алармима, и другим врстама порука, и омогућава једноставну интеграцију са другим AWS сервисима за аутоматизацију процеса.

### 2.2.8 Amazon EventBridge

*Amazon EventBridge* [10] је сервис за управљање догађајима који омогућава међусобно увезивање апликација и сервиса, заказивање извршавања одређених акција и слично. EventBridge омогућава креирање правила за филтрирање и усмеравање догађаја, што олакшава изградњу реактивних и скалабилних апликација, и често се користи за оркестрацију догађаја између различитих сервиса и апликација

### 2.2.9 AWS CloudWatch

*AWS CloudWatch* [11] је сервис за праћење, надгледање и управљање ресурсима и апликацијама на AWS-у. Омогућава конфигурацију, прикупљање и праћење различитих врста логова, метрика и догађаја из AWS сервиса. Такође, омогућава и креирање аларма и визуелних приказа за праћење перформанси и стања апликације, што олакшава идентификовање и решавање евентуалних проблема. Нуди и могућност аутоматизованих одговора на одређене догађаје, што на пример може бити покретање Lambda функција као одговор на одређене метрике. CloudWatch представља основни сервис за надгледање употребе ресурса и перформанси апликација на AWS-у.

### 2.2.10 AWS IAM (Identity and Access Management)

*AWS IAM* [12] је сервис за контролу приступа AWS ресурсима. Омогућава дефинисање полиса и њихово додељивање ролама, на основу којих се одређује који сервиси и под којим условима имају право приступа другим ресурсима и сервисима у оквиру AWS-а. Омогућава грануларну контролу приступа, а користи се и за управљање корисничким дозволама, креирање и управљање корисничким налозима и групама, и генерално обезбеђивање сигурног приступа ресурсима

## 2.3. Node.js

*Node.js* [13] је радно окружење за извршавање *JavaScript* кода на серверској страни, односно ван веб прегледача. Базиран је на V8 JavaScript engine-у који користи Google Chrome претраживач. Захваљујући својим асинхроним и догађајима управљаним (*event-driven*) карактеристикама, погодан је за изградњу апликација које захтевају паралелно извршавања или real-time обраду података. Још једна важна карактеристика је lightweight природа Node.js-а, која га чини идеалним избором за serverless архитектуру и коришћење у AWS Lambda функцијама.

## 2.4. Angular

*Angular* [14] је компонентно оријентисан радни оквир за развој скалабилних веб апликација. Комплетно је написан у TypeScript-у. Користи се за изградњу клијентског дела апликација користећи HTML [26] и TypeScript. Омогућава креирање динамичких, једностраних апликација (Single Page Applications - SPA) које су брзе, интерактивне и лаке за одржавање.

## 3. СПЕЦИФИКАЦИЈА ПРОЈЕКТА

### 3.1. Преглед пројекта

CloudCast представља сервис базиран на принципу претплате који корисницима омогућава да добијају персонализована обавештења о временским прилика-кама.

Након што се успешно региструје, а самим тим и претплати на услуге сервиса, корисник ће добијати дневна обавештења у складу са конфигурацијом налога. Осим тога, има могућност пријаве на сам сервис путем свог корисничког налога у оквиру ког може да промени конфигурацију претплате, евентуално се одјави са претплате, као и да прегледа тренутне временске услове и прогнозу за наредних пет дана за град који је специфицирао.

Поред крајњих корисника система, препозната је и улога администратора. Администратор на недељном нивоу путем мејла добија дијаграме који представљају информације извучене из тренутно активних претплата.

Ови дијаграми представљају тренутне односе различитих типова претплата, градова које су корисници специфицирали у оквиру претплате и слично. Такође, све ове информације се чувају и у оквиру система како би администратор могао да прегледа историјске податке и анализира тренд промена у подацима

### 3.2. Функционални захтеви

Систем препознајте три врсте корисника, и следеће функционалне захтеве у оквиру сваке категорије корисника:

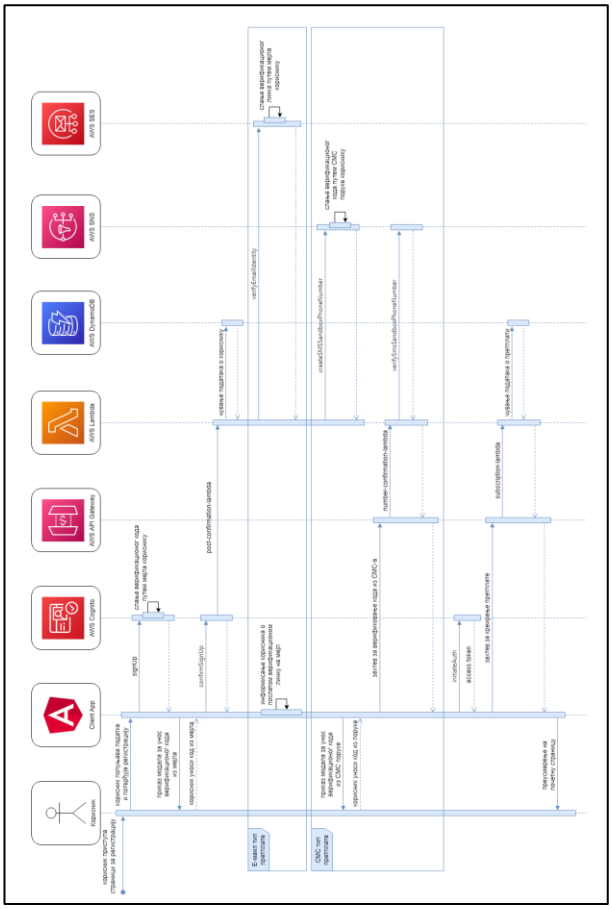
- Неулоговани корисник
  - Пријава на систем
  - Регистрација
- Регуларни корисник
  - Добијање дневних обавештења о временској прогнози у складу са конфигурацијом претплате
  - Промена статуса претплате
  - Промена конфигурације претплате
  - Преглед тренутних временских услова и прогнозе за наредних 5 дана
- Администратор
  - Добијање недељних извештаја о броју активних претплата као и дијаграма који приказују расподеле претплата по различитим конфигурационим параметрима
  - Преглед историјских података

### 3.3. Архитектура система

Архитектура сервиса CloudCast заснована је на serverless приступу и дефинисана је као инфраструктура као код (IaC) користећи Terraform. Овај приступ омогућава аутоматско креирање и управљање ресурсима у AWS окружењу. Клијентска апликација је развијена помоћу Angular-a, пословна логика апликације у Node.js-у.

### 3.4. Дијаграми

Неке од комплекснијих функционалности у систему представљени су дијаграмима секвенци ради лакшег разумевања тока догађаја. У наставку представљен је дијаграм секвенци за процес регистрације (слика 1).

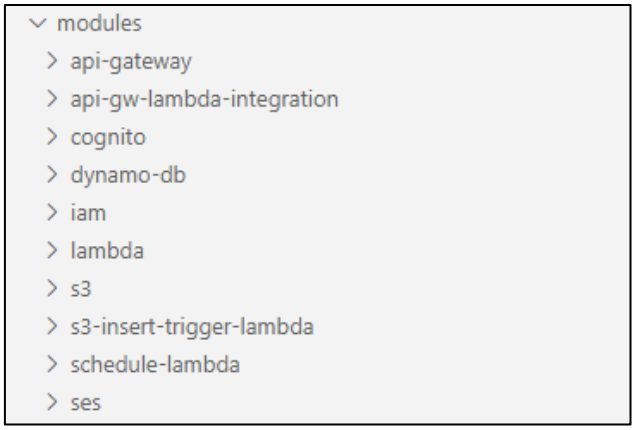


Слика 1. *Дијаграм секвенци за функционалност регистрације*

## 4. ИМПЛЕМЕНТАЦИЈА СИСТЕМА

### 4.1. Инфраструктура

Комплетна инфраструктура налази се на AWS-у, а за подизање инфраструктуре коришћен је Terraform IaC алат. Код инфраструктуре организован је у модуле који служе за груписање повезаних ресурса и омогућују једноставан начин за њихову вишеструку употребу и одржавање.




Слика 2. *Преглед свих модула у пројекту*

Сваки модул састоји се из три фајла: *main.tf* у оквиру ког се ддекларишу terraform ресурси, а на основу којих се креирају одговарајући ресурси у оквиру инфраструктуре, *outputs.tf* фајл у оквиру ког се декларишу output terraform параметри и *variables.tf* у оквиру ког се декларишу variable terraform параметри, помоћу којих је могуће конфигурисати одређене делове модула на месту њихове декларације.

### 4.2. Бизнис логика апликације

Целокупна бизнис логика сервиса налази је организована у осам lambda функција.

### 4.3. Клијентска апликација

За имплементацију клијентске апликације коришћен је радни оквир *Angular*. Клијентска апликација састоји се од 6 компоненти и *servis* директоријума. Од ових 6 компоненти, при чему прве 4 у наставку представљају читаве странице, а уједно и све странице које постоје у оквиру клијентске апликације, док преостале две представљају модалне прозоре који се појављују у оквиру тих страница

## 5. ЗАКЉУЧАК

CloudCast представља сервис за слање обавештења о временским приликама за одређени град. Идеја за овај рад настала је услед недостатка сличних решења на тржишту. Предност овог сервиса у односу на већ постојећа, слична решења је претплатнички механизам, као и могућност примања обавештења путем СМС порука. У претходним поглављима поменути су најпопуларнија слична решења на тржишту, наведене су технологије коришћене у имплементацији сервиса, спецификација система укључујући и UML дијаграме, детаљи имплементације и на крају демонстрација рада сервиса.

Када је реч о даљем развоју функционалности сервиса, један од могућих праваца могао би бити проширење информација о временским приликама које се шаљу кориснику. Такође, потенцијална нова функционалност је неки вид аларма који би послао упозорење свим преплатницима одређеног града уколико је велика временска непогода најављена за ту регију. Са друге стране, што се тиче инфраструктурног дела система и техничке имплементације, потенцијална унапређења могла би се тражити у проналажењу serverless решења и за клијентску апликацију, како би била у складу са остатком инфраструктуре.

## 6. ЛИТЕРАТУРА


[1] What is Terraform | Terraform | HashiCorp Developer. [Online], Приступљено датума: 22.8.2024. https://developer.hashicorp.com/terraform/intro

[2] About AWS. [Online], Приступљено датума: 22.8.2024. https://aws.amazon.com/about-aws/

[3] Amazon API Gateway. [Online], Приступљено датума: 22.8.2024. https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html

[4] Amazon Cognito. [Online], Приступљено датума: 22.8.2024. https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html

[5] AWS Lambda. [Online], Приступљено датума: 23.8.2024. https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

[6] Amazon DynamoDB. [Online], Приступљено датума: 22.8.2024. https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html

[7] Amazon Simple Storage Service. [Online], Приступљено датума: 22.8.2024. https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html

[8] Amazon Simple Email Service. [Online], Приступљено датума: 22.8.2024. https://docs.aws.amazon.com/ses/latest/dg/Welcome.html

[9] Amazon Simple Notification Service. [Online], Приступљено датума: 22.8.2024. https://docs.aws.amazon.com/sns/latest/dg/welcome.html

[10] Amazon EventBridge. [Online], Приступљено датума: 22.8.2024. https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-what-is.html

[11] Amazon CloudWatch. [Online], Приступљено датума: 22.8.2024. https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html

[12] AWS Identity and Access Management. [Online], Приступљено датума: 22.8.2024. https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

[13] Node.js — About Node.js. [Online], Приступљено датума: 22.8.2024. https://nodejs.org/en/about

[14] Angular - What is Angular? [Online], Приступљено датума: 22.8.2024. https://v17.angular.io/guide/what-is-angular


**Кратка биографија:**


**Ђорђе Његић** рођен је 5. септембра 1999. године у Суботици где је стекао основно и средње образовање. Школске 2018/2019 године уписује се на Факултет техничких наука Универзитета у Новом Саду, смер Софтверско инжењерство и информационе технологије. Основне академске студије завршио је 2022. године и исте године уписује мастер академске студије на истом студијском програму. Контакт: djnjegic@gmail.com