

Računalniške Storitve v Oblaku - Poročilo

Extreme Weather Event Notifier

Projektna ekipa 19, Jaka Škerjanc in Erik Pahor

2025

Povezava do repozitorija kode

Projektna koda je dostopna na javno dostopnem repozitoriju GitHub, kar omogoča vpogled v implementacijo ter CI/CD konfiguracijo:

- Repozitorij: <https://github.com/rso2425/extreme-weather-event-notifier>
- Aplikacija: <https://rso-weather.duckdns.org/>

Kratek opis projekta

Extreme Weather Event Notifier je spletna aplikacija, ki uporabnike obvešča o ekstremnih vremenskih dogodkih v Sloveniji. Uporabniki lahko omogočijo prejemanje potisnih obvestil in dostopajo do zgodovine vremenskih dogodkov. Aplikacija zbira podatke iz Agencije Republike Slovenije za okolje (ARSO) in jih obdeluje v mikrostoritvenem sistemu. Potisna obvestila se pošiljajo preko Firebase Cloud Messaging, podatki pa se shranjujejo v MongoDB in so dostopni preko spletnega vmesnika.

Ogrodje in razvojno okolje

Pri razvoju projekta so bili uporabljeni sodobni jeziki in orodja, ki omogočajo skalabilno in vzdrževalno infrastrukturo: Frontend je implementiran s pomočjo Vue.js, za backend pa se uporablja Express.js. Podatki se shranjujejo v MongoDB, medtem ko RabbitMQ služi za medstoritveno komunikacijo. Infrastruktura temelji na Dockerju in Kubernetesu, CI/CD procesi pa so avtomatizirani s pomočjo GitHub Actions.

Arhitektura sistema

Arhitektura aplikacije temelji na mikrororitvenem modelu, kjer vsaka storitev izpolnjuje določeno funkcijo:

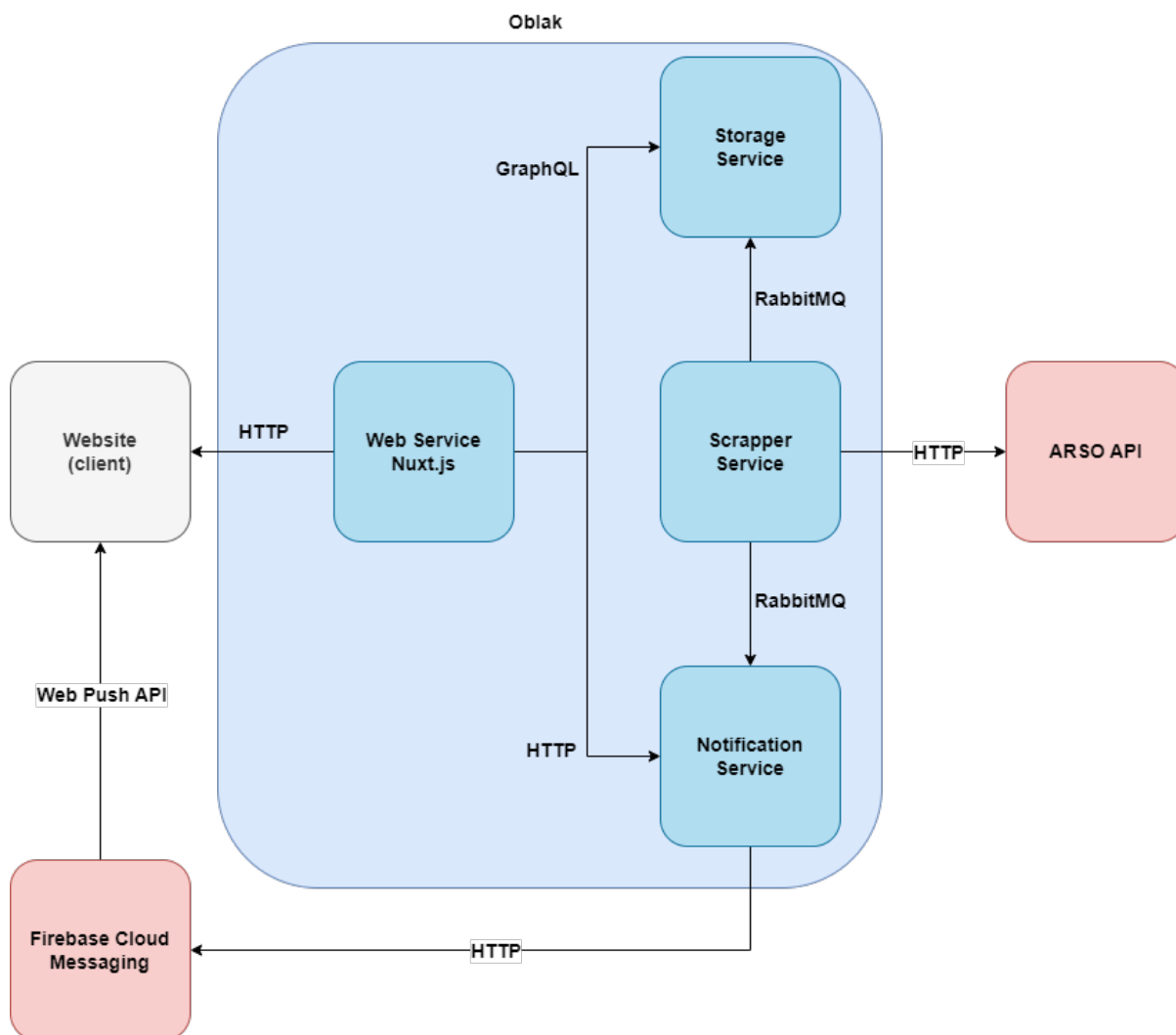


Figure 1: Shema arhitekture aplikacije

Aplikacija sestavljajo štiri glavne mikrororitve: Web (frontend), Notification, Scraper in Storage. Komunikacija med storitvami poteka preko RabbitMQ in API-jev (REST in GraphQL). Push obvestila se pošiljajo preko Firebase Cloud Messaging.

Opis mikrororitv

Web Mikrororitev (Frontend)

Web mikrororitev predstavlja uporabniški vmesnik aplikacije, implementiran v Vue.js. Uporabnikom omogoča:

- Pregled zgodovine vremenskih dogodkov, ki se pridobijo iz Storage mikrororitve preko GraphQL.
- Prijavo in odjavo na potisna obvestila preko Notification mikrororitve.
- Upravljanje potisnih obvestil s Firebase Cloud Messaging.

Notification Mikrostoritev

Notification mikrostoritev je odgovorna za pošiljanje potisnih obvestil in upravljanje Firebase ID-jev uporabnikov. Prejema dogodke iz Scraper mikrostoritve preko RabbitMQ in pošilja obvestila uporabnikom z registriranimi napravami.

Scraper Mikrostoritev

Scraper mikrostoritev periodično zbira podatke o vremenskih dogodkih iz ARSO preko CAP XML standarda. Ti podatki se obdelujejo in pošiljajo Notification ter Storage mikrostoritvama preko RabbitMQ.

Storage Mikrostoritev

Storage mikrostoritev shranjuje podatke o vremenskih dogodkih v MongoDB in izpostavlja GraphQL API, ki omogoča pridobivanje podatkov za prikaz zgodovine dogodkov v Web mikrostoritvi.

CI/CD in testiranje

CI/CD proces je implementiran s pomočjo GitHub Actions. Vključuje:

- Zaščito glavne veje "main", kjer so spremembe možne le preko pull requestov.
- Avtomatizirano gradnjo in testiranje Docker slik.
- Nameščanje posodobljenih slik v Kubernetes preko DigitalOcean Container Registry.

Testiranje je razdeljeno na unit teste za posamezne storitve, integracijske teste za preverjanje komunikacije med storitvami in ročno testiranje uporabniškega vmesnika.

Primeri uporabe

Aplikacija omogoča uporabniku enostavno interakcijo: Uporabnik obišče spletno mesto, omogoči potisna obvestila, in prejme Firebase ID, ki je shranjen v Notification mikrostoritvi. Scraper zazna vremenski dogodek, ki se pošlje v RabbitMQ. Notification pošlje obvestilo uporabnikom, Storage pa omogoča vpogled v zgodovino dogodkov.

Seznam opravljenih projektnih zahtev

• Repozitorij

- Ustvarjen GitHub repozitorij dostopen na [rso2425/extreme-weather-event-notifier](https://github.com/rso2425/extreme-weather-event-notifier).
- Ustvarjena je datoteka README.md, ki opisuje projekt (README.md).
- Dodana je omejitev na glavno vejo, spremembe so možne le preko pull requestov in razvijalske veje.

• Mikrostoritve in »cloud-native« aplikacija

- Osnova struktura projekta in razvojnega okolja je vidna v repozitoriju.
- Vsaka mikrostoritev ima implementirana navodila za namestitev in zagon.
- Storitve komunicirajo prek HTTP in RabbitMQ sporočilnega protokola.
- Vsaka storitev ima svoje Kubernetes manifeste.
- Uporabljena je MongoDB NoSQL podatkovna baza, z nekaj podatkih o vremenskih dogodkih.

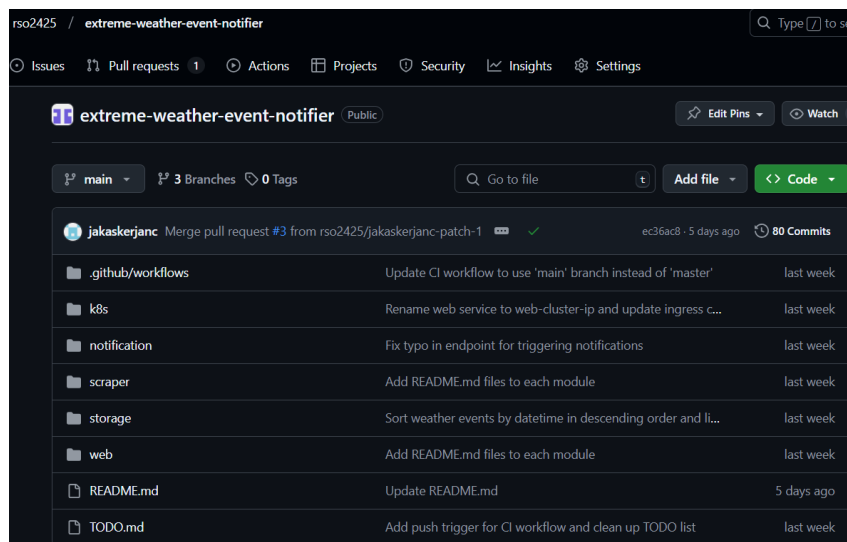


Figure 2: Struktura repozitorija

- **Dokumentacija**

- Dodan dokument README.md, ki podrobno opisuje projekt. (README.md)

- **Dokumentacija API**

- Javno dostopne končne točke API-jev so dokumentirane z OpenAPI specifikacijo. (Dokumentacija)
- Dodani so primeri zahtev in odgovorov.

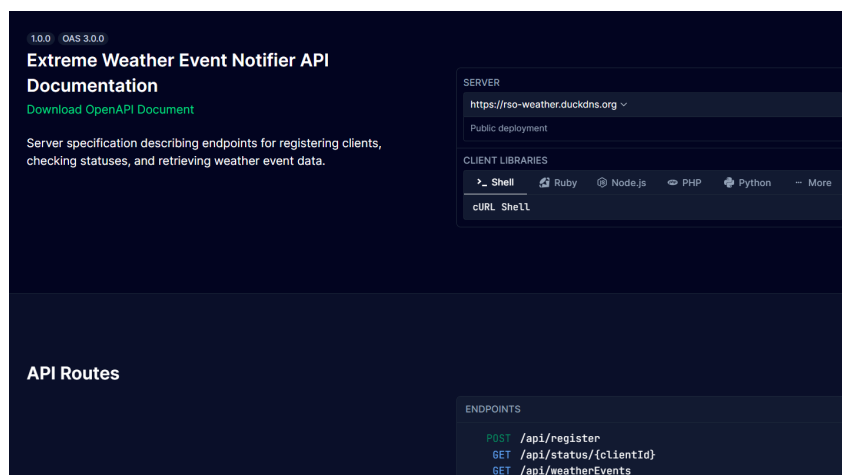


Figure 3: Dokumentacija API

- **Cevovod CI/CD**

- Ustvarjen je cevovod CI/CD, ki izvede gradnjo Docker slik, testiranje in namestitve v DigitalOcean.
- Uporabljen je GitHub Actions za izvedbo cevovoda.
- Ustvarjena je datoteka .github/workflows/ci.yaml, ki opisuje cevovod. (ci.yaml)

- **Namestitve v oblak**

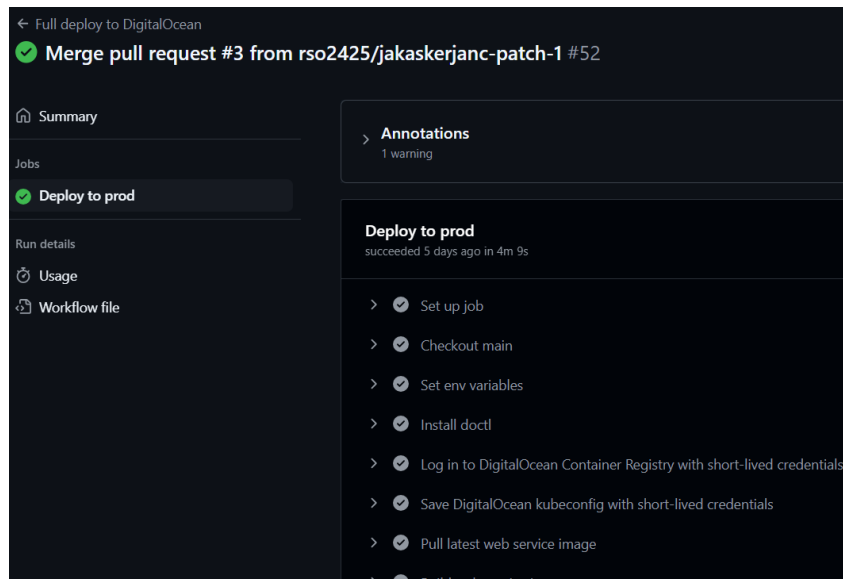


Figure 4: Cevovod CI/CD

- Ustvarjena je infrastruktura v DigitalOcean.
- Namestitev je javno dostopna na rso-weather.duckdns.org.

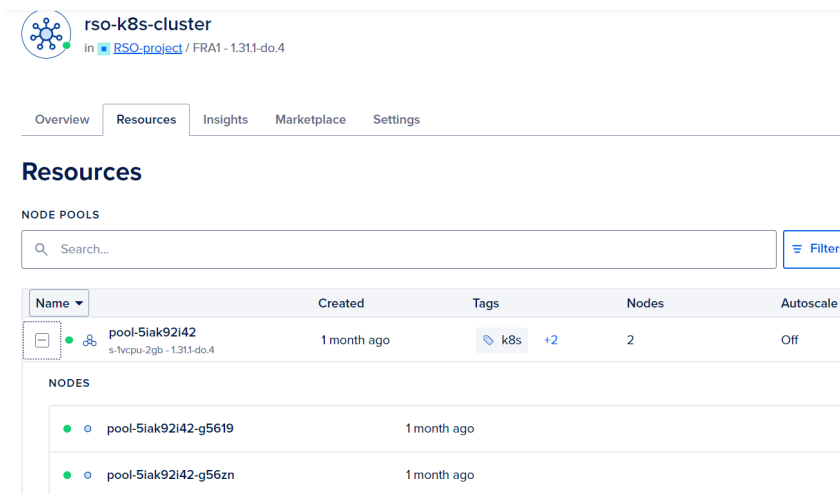


Figure 5: Infrastruktura v oblaku

• Zunanji API

- Dodan je zunanji API, ki omogoča pridobivanje podatkov o vremenskih dogodkih.
- ARSO API je javno dostopen zato autentikacija ni potrebna.
- Uporabljano je Firebase Cloud Messaging za pošiljanje potisnih obvestil.
- Privatni ključi niso shranjeni v repozitoriju ampak se prenesejo v Docker sliko ob gradnji.

• Preverjanje zdravja

- Vsaki storitvi je dodana končna točka '/healthz', ki preverja zdravje aplikacije.
- Kubernetes manifest je konfiguriran tako, da preverja zdravje storitev.

```

const sloveniaArsoLinks = {
  southEast: 'https://meteo.arso.gov.si/uploads/probase/www/warning/text/sl/warning_SLOVENIA_SOUTH-EAST_latest_CAP.xml',
  southWest: 'https://meteo.arso.gov.si/uploads/probase/www/warning/text/sl/warning_SLOVENIA_SOUTH-WEST_latest_CAP.xml',
  middle: 'https://meteo.arso.gov.si/uploads/probase/www/warning/text/sl/warning_SLOVENIA_MIDDLE_latest_CAP.xml',
  northEast: 'https://meteo.arso.gov.si/uploads/probase/www/warning/text/sl/warning_SLOVENIA_NORTH-EAST_latest_CAP.xml',
  northWest: 'https://meteo.arso.gov.si/uploads/probase/www/warning/text/sl/warning_SLOVENIA_NORTH-WEST_latest_CAP.xml'
}

async function fetchCAPData(url: string) {
  try {
    const response = await fetch(url)
    const xml = await response.text()
    const cap = await CAP_1_2.Alert.fromXML(xml)
    return cap
  } catch (error) {
    console.error('Error fetching or parsing data:', error)
  }
}

```

Figure 6: Zunanji API

- GraphQL

- Dodana je GraphQL podpora v Web mikrostoritvi.
- Dodana je GraphQL podpora v Notification mikrostoritvi.

```

const schema = buildSchema(`
  type WeatherEvent {
    id: String!
    datetime: String!
    title: String!
    severity: Int!
    description: String!
    instruction: String!
    region: String!
    source: WeatherEventSource!
    type: WeatherEventType!
  }

  enum WeatherEventType {
    RAIN
    SNOW
    WIND
    OTHER
  }

  enum WeatherEventSource {
    ARSO
  }

  type Query {
    weatherEvents: [WeatherEvent!]!
    weatherEvent(id: String!): WeatherEvent
  }
`)

```

Figure 7: GraphQL Shema

- **Sporočilni sistem**

- Dodan je RabbitMQ sporočilni sistem.
- Mikrostoritve Notification, Scraper in Storage komunicirajo preko RabbitMQ.

```
const openChannel = async () => {
  try {
    console.log('Connecting to RabbitMQ server...')
    const connection = await amqp.connect('amqp://rabbitmq')
    const channel = await connection.createChannel()
    console.log('Channel opened')
    return { connection, channel }
  } catch (error) {
    console.error('Error opening channel:', error)
    throw error
  }
}

const sendMessage = async (channel: amqp.Channel, queue: string, message: object) => {
  try {
    // Ensure the queue exists
    await channel.assertQueue(queue, {
      durable: true // Make queue persistent
    })

    const messageBuffer = Buffer.from(JSON.stringify(message))

    // Send the message to the queue
    channel.sendToQueue(queue, messageBuffer, {
      persistent: true // Make message persistent
    })

    console.log(`Sent: ${JSON.stringify(message)}`)
  } catch (error) {
    console.error('Error sending message:', error)
    throw error
  }
}
```

Figure 8: RabbitMQ koda

- **Grafični vmesnik**

- Dodan je grafični vmesnik v Web mikrostoritvi.
- Vsebuje več podstrani.
- Za dostop do podatkov uporablja REST API.

Extreme Weather Event Notifier

List of all previous weather events in Slovenia

Moderate Wind Warning Severity: 2 Date and time: 11/01/2025 06:59 Description: Maximum wind speed : from 50 to 70 km/h. Wind sways trees and may break smaller branches. Instruction: Doors and windows should be kept closed. We recommend additional caution in traffic, especially in crosswinds and in exposed areas (e.g. bridges, cuttings). Do not linger in potentially dangerous areas. Region: Slovenia / North-East Source: ARSO	Moderate Snow Avalanches Warning Severity: 2 Date and time: 11/01/2025 06:59 Description: The snowpack is only moderately well bonded on some steep slopes, otherwise well bonded in general. Snow avalanche risk level - MEDIUM - (according to 5 levels of the European avalanche risk table) LEVEL : 3 *. Triggering is possible with high additional loads, particularly on the steep slopes indicated in the bulletin. Large natural avalanches not likely. Instruction: Heightened avalanche conditions on specific terrain features. Evaluate snow and terrain carefully; identify features of concern. Region: Slovenia / North-West Source: ARSO
---	---

Figure 9: Grafični vmesnik

- **Ingress Controller**

- Dodan je Ingress Controller, ki omogoča javni dostop do aplikacije.
- Implementiran je TLS/SSL samopodpisani certifikat.

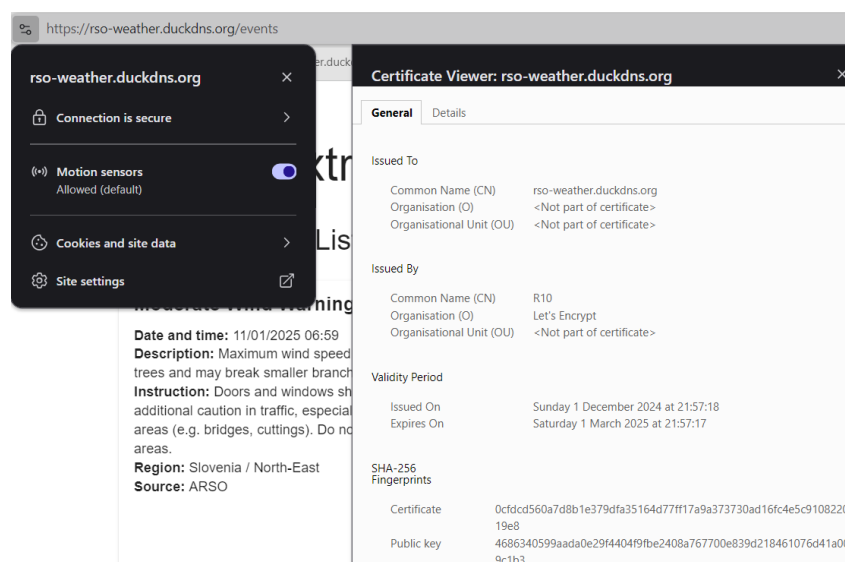


Figure 10: SSL certifikat