Versione 20.9.2022 (Middle)

Inglese

Description

Contained in this repo, there are some instructions for a new application that will go live in the next month!

You will need to:

1. Create a new git public GitHub (or similar services) repository and work there.
2. Automate the creation of the infrastructure and the setup of the application.

Requirements

You have these instructions:

3. It's based on the last version of WordPress (it will be more useful if we can parameterize the version)
4. You can choose Apache, Nginx, or whatever you want
5. Once deployed, the application should be:
- Secure
- Fast
- Fault-tolerant
- Adaptive to average load
6. You can use whatever tool you want to provision the infrastructure (CloudFormation, Terraform, CDK)
7. **Optional: Create a CI/CD pipeline to deploy WordPress**
8. Write a readme with an architecture diagram and all the required instructions to install and try your solution.

Make any assumptions that you need to. This is an opportunity to showcase your skills, so if you want to, implement the deployment process with any additional features, tools, or techniques you'd like to.

We are evaluating solutions based on the architecture and quality of the deployment. Show us just how beautiful, clean and pragmatic your code can be.

Once your solution is ready, please send us the link to your project.

Italiano

Descrizione

In questo repo ci sono delle istruzioni per creare una nuova applicazione che andrà live il mese prossimo!

Per raggiungere l'obiettivo, avrai bisogno di:

9. Creare un nuovo repository git pubblico su GitHub (o servizi simili) su cui lavorare.
10. Automatizzare la creazione dell'infrastruttura e setup dell'applicazione.

Requisiti

I requisiti sono i seguenti:

1. L'applicativo è basato sull'ultima versione di WordPress (sarebbe comunque indicato poter parametrizzare la versione)
2. Puoi scegliere Apache, Nginx, o quello che preferisci
3. Una volta deployato, l'applicativo dovrebbe essere:

- Sicura
- Veloce
- Fault-tolerant
- Scalabile in base al carico medio

4. Puoi usare lo strumento che preferisci per il provisioning dell'infrastruttura (CloudFormation, Terraform, CDK)
5. **Opzionale: Creare una pipeline di CI/CD per il deploy di WordPress**
6. Scrivere un readme comprensivo di un diagramma infrastrutturale e di tutte le istruzioni necessarie per installare e provare la tua soluzione.

Fai qualsiasi assunzione di cui hai bisogno. Questa è una opportunità per dimostrare le tue abilità quindi, se vuoi, implementa tutte le funzionalità ed utilizza gli strumenti o le tecniche che desideri.

Valutiamo le soluzioni basandoci sulla architettura e sulla qualità del deployment. Dimostraci quanto elegante, pulito e pragmatico può essere il tuo codice.

Quando hai terminato la tua soluzione, mandaci per favore il link del tuo progetto.

Versione 20.9.2022 (Senior)

Phoenix Application Problem

Inglese

This problem is about creating a production ready infrastructure for the Phoenix Application.

Problem

The development team has released the phoenix application code (you can find the codebase here). Your task is to create the production infrastructure for the Phoenix

application. You must pay attention to some unwanted features that were introduced during development. In particular:

- GET /crash kill the application process
- GET /generatecert is not optimized and creates resource consumption peaks

General Requirements

- You may use whatever programming language/platform you prefer. Use something that you know well.
- You must release your work with an OSI-approved open source license of your choice.
- You must deliver the sources, with a README that explains how to run it.
- Add the code to your own Github account and send us the link.

Application Requirements

- Runs on Node.js 8.11.1 LTS
- MongoDB as Database
- Environment variables:
    - PORT - Application HTTP Exposed Port
    - DB_CONNECTION_STRING - Database connection string mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]

Run Application

- Install dependencies npm install
- Run npm start
- Connect to http://<hostname|IP>:<ENV.PORT>

Problem Requirements

1. Automate the creation of the infrastructure and the setup of the application.
2. Recover from crashes. Implement a method auto restart the service on crash
3. Backup the logs and database with rotation of 7 days
4. Notify any CPU peak
5. Implements a CI/CD pipeline for the code
6. Scale when the number of request are greater than 10 req /sec

Italiano

È necessario creare una infrastruttura di produzione per il rilascio della Applicazione Phoenix.

Problema

Il team di sviluppo ha rilasciato il codice della applicazione phoenix (puoi trovare la codebase qui). Il tuo compito è quello di creare l'infrastruttura di produzione per

l’applicazione Phoenix. Dovrai fare attenzione ad alcune funzioni indesiderate che sono state introdotte durante lo sviluppo. In particolare:

- GET /crash kill del processo dell’applicazione
- GET /generatecert non è ottimizzato e crea picchi nel consumo di risorse

Requisiti generali

- Puoi usare qualsiasi linguaggio di programmazione/piattaforma che preferisci. Usa quello che conosci bene.
- Rilascia il tuo lavoro con una licenza open source OSI-approved di tua scelta.
- Dovrai consegnare i sorgenti con un README che spiega come va eseguito il codice.
- Aggiungi il codice al tuo account GitHub e mandaci il link una volta finito.

Requisiti applicativi

- Gira su Node.js 8.11.1 LTS
- Il Database è MongoDB
- Variabili d’ambiente:
    - PORT - Porta HTTP esposta dall’applicativo
    - DB_CONNECTION_STRING - Stringa di connessione al database mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database][?options]]

Far partire l’applicazione

- Installa le dipendenze con npm install
- Esegui npm start
- Connettiti a http://<hostname|IP>:<ENV.PORT>

Requisiti del problema

1. Automatizzare la creazione dell’infrastruttura ed esegui l’installazione dell’applicazione.
2. Recupera dopo eventuali crash. Implementa un metodo di riavvio automatico del servizio in caso di crash
3. Eseguire il backup dei log e del database con una rotazione d 7 giorni
4. Notifica eventuali picchi di utilizzo della CPU
5. Implementa una pipeline di CI/CD per il codice
6. Scala quando il numero di richieste è maggiore di 10 richieste al secondo