DevOps Challenge

Introduction

This challenge to know more about your skills and style.

The Challenge

This summer is too hot in Oman, we want to make a web API to get the weather forecast, and the API should have 2 endpoints `/weather` and `/version`.

You are free to design the solution that should be able to handle hundreds of millions of requests. This app has a RESTful API, so output should be in JSON.

`/weather` endpoint

Parameters:

- City and country name or latitude and longitude.
- Temperature warning threshold.
- Temperature unit (Metric or Imperial).
- Wind warning threshold.
- Wind unit (Metric or Imperial).

- The temperature, precipitation, humidity, and wind of today's and next 2 days.
- Split the day to 3 sections, morning, midday, and night.
- Warning if the weather passed a specific threshold.
- The right gadget for that weather (e.g. umbrella, sunglass, raincoat).

`/version` endpoint

Parameters:

- Just call the endpoint with no parameters.

Response:

- Current version of the app.

Mandatory requirements

The following criteria is mandatory:

- Documentation
- The endpoints should work as described
- Automated build and CI pipeline
- Zero downtime deployment

Optional/Plus requirements

- Using containers
- Unit test
- Caching
- Monitoring endpoint that gives the app status
- A to-do list of what could be enhanced later

What are we looking for?

We are looking for the balance between the Dev and Ops in the `DevOps`! So we are interested in the code as well architecture (scalability) and foundation (deployment style, logging, configuration, security, scalability, development environment, etc), than on peculiarities of the business logic.

Guidelines:

- You can use either Go, Python or Java.
- We value simple solutions over anything complex. Don't over-engineer.
- Estimating the time needed for completing the task is part of the task itself. After reading this document, please assess how much time you'll need and please notify the recruiter.
- This doesn't need to be production-ready code, but be prepared to explain what further changes you'd make if you had more time.
- Automate everything, from the build process to the deployment. A suggestion is to use Vagrant with other DevOps tools to keep the host clean.
- The system must be able to run on Linux, MacOS, and Windows. You can support other systems too if you wish.
- You can use any external components, like database systems, if you wish, but remember to automate.
- Please write documentation on how to use the system and place it in the repository.
- Aslo write about your architectural choices in the doc.
- Please, create a private Git repository (Bitbucket, Github or Gitlab - your choice!) and share the access with us. After you choose where you'll host it we can send you our usernames.
- We will check your commits to see your progress in the project, so, push them as they are committed.
- Please, let the recruiter know when you have finished. Make sure to hand in your results by the agreed deadline, and send us the repository URL.