Technical Interview Exercise

Your task is to deploy a sample API with potentially several replicas and its database using persistent storage. Ideally you will run the Azure Kubernetes Service as orchestrator of the deployment while adhering to the best practices in the technology.

You must allow these resources to communicate with each other and that the API is accessible from the outside, while the database is not. To define, install and manage your applications, ideally you will use Helm.

To showcase your ability to work with modern systems you cannot generate these resources manually, instead you should be focusing on IaC technologies like Terraform.

Your deployment should be driven by an informal user story that you will create, and which should be included in your presentation.

The deployment should include the following components

Terraform code: Create the resources necessary to deploy this infrastructure using a repository structure that considers potential multiple environments and an organizational approach.

Helm charts: Charts describing all cluster resource sets where applicable. Kustomize is an open option here.

Dockerfiles: Sample Dockerfiles images implemented.

User story: A simple epic listing the tasks with which you would organize this deployment.

The resources can remain on **standby** (git repository). Running this infrastructure in Azure is a plus. Implementing a CI/CD system to automate deployment is a plus. GitOps is a plus. Any additional component that improves the complexity of this exercise is welcome: DevOps/SRE practices, Monitoring, Log Management, Alerts, etc.

Presentation and Code Review

Once you have completed the exercise, you will be required to present your project to the technical interview panel. During the presentation, you should explain your user story, design choices, the technical architecture, and as an **optional** demonstrate the functionality of the deployment. This will be done over Zoom and you will screen share either your GitHub repository or IDE. This Github repository must be used to share the exercise in advance at the time requested.

After the presentation, the interview panel will conduct a code review of your project. You will be asked to explain your coding decisions and answer any questions related to the code. The interview panel will evaluate your project based on the following criteria:

Terraform skills: Ability to handle the tool. Level of organization, best practices and long-term mindset. Skills to fully utilize the potential of IaC.

Kubernetes skills: Ability to handle the tool. Technical considerations for the creation of resources. Use of management tools for definition, installation and management.

Code quality: Your code should be well-organized, readable, and adhere to best practices.

Functionality: Your application should behave as expected, it should be accessible as described and any basic modifications to the base code should make the expected effect. Keep in mind that deployment to Azure Cloud is **optional**, so the technical decisions and relations between resources must be clearly explained in the review.

User Story: Your user story should drive the deployment of the application and be included in your presentation.

Presentation: Your presentation should be clear, concise, and demonstrate a good understanding of the project.

Scoring

You will be scored based on your adherence to the best practices in Kubernetes and Terraform, management tools usage, code quality, user story, and code review remarks. Each criterion will be weighted equally. The maximum score for the exercise is 100 points.

We hope this exercise challenges your technical abilities and helps us assess your suitability for the intermediate full-stack software developer position. Good luck!