SAVANNAH Informatics

5th Floor,
One Padmore Place,
off George Padmore
Road. Nairobi, Kenya
P.O. Box 5980-00200.

Backend Technical Challenge:

Background

Most of our API services are written in Python and Go:

- REST and GraphQL APIs written in Python/Django/Django REST Framework and Go
- Majority of our services are containerized and running on Kubernetes. Kubernetes deployments are orchestrated using Helm charts. We also employ Ansible playbooks for configuration management for deployment on Debian-derivative Linux servers.
- PostgreSQL as the database of choice; although we don't shy away from using Oracle or SQL Server when a client's requirements demand so

The technical interview will be built around a coding assignment that is designed to screen for the following basic skills:

- Experience in developing REST and GraphQL APIs in Python/Go
- Experience with a configuration management tool e.g., Chef, Puppet, Ansible etc. Experience working with an infrastructure as code tool e.g. Terraform or Pulumi will be an advantage.
- Experience working with containers and container orchestration tools e.g. k8s
- Experience writing **automated tests** at all levels - unit, integration, and acceptance testing
- Experience with **CI/CD** (any CI/CD platform)

We will be particularly interested in:

- Testing + coverage + CI/CD
- HTTP and APIs e.g., REST
- OAuth2
- Web security e.g., XSS
- Logic / flow of thought
- Setting up a database
- Version control

+254 790 360 360 | info@savannahinformatics.com | www.savannahinformatics.com

SAVANNAH Informatics

5th Floor,
One Padmore Place,
off George Padmore
Road. Nairobi, Kenya
P.O. Box 5980-00200.

Screening Test

1. Create a simple Python or Go service
2. Design a simple customers and orders database (keep it simple)
3. Add a REST or GraphQL API to input / upload customers and orders:
   - Customers have simple details e.g., name and code.
   - Orders have simple details e.g., item, amount, and time.
4. Implement authentication and authorization via OpenID Connect
5. When an order is added, send the customer an SMS alerting them (you can use the Africa's Talking SMS gateway and sandbox)
6. Write unit tests (with coverage checking) and set up CI + automated CD. You can deploy to any PAAS/FAAS/IAAS of your choice
7. Write a README for the project and host it on your GitHub