O11Y Test Task

Context:

Veriff has a mesh of Kubernetes environments. You can explore the facade by visiting https://demo.veriff.me. The task focuses on creating environments, running applications unknown to you, and securing them. It's also essential to document your work so that colleagues can pick up where you left off.

Objective:

Demonstrate your skills in Kubernetes, observability (o11y), and application deployment by completing the following task.

Task Requirements:

1. Create a Kubernetes Cluster:
    a. Use any technology of your preference (e.g., Minikube, Kind, K3s, EKS, GKE, AKS, etc.).
    b. Ensure the cluster is configured to support deployments and observability tooling.
2. Deploy a Telemetry Collection Stack:
    a. Choose an observability stack (e.g., ELK, LGTM, OpenTelemetry, Prometheus + Grafana, etc.).
    b. Ensure the stack collects and stores logs, metrics, and traces.
3. Deploy an Application:
    a. Deploy any application of your choice (e.g., a simple web app, an API service, etc.).
    b. The application should have observable telemetry (logs, metrics, traces).
4. Monitor the Service:
    a. Implement monitoring for the deployed application.
    b. Demonstrate the ability to identify key metrics and set up at least one alert based on observed telemetry.

O11Y Test Task

Deliverables:

1. Observability Solution Design:
   a. Provide a detailed explanation of your chosen observability stack.
   b. Describe how it meets the task requirements and why you chose it.
2. Telemetry Collection:
   a. Demonstrate logs, metrics, and traces being collected from all layers (infrastructure, network, application).
   b. Include visualizations (e.g., screenshots of dashboards) to showcase the data.
3. Monitoring Configuration:
   a. Show how the stack monitors the infrastructure, network, and application.
   b. Highlight key metrics and configured alerts.
4. Documentation:
   a. Provide detailed instructions on how you completed the task.
   b. Include architectural diagrams, configurations, and troubleshooting approaches.
   c. Ensure the documentation is clear and reproducible.

Assessment Criteria:

1. Technical Completeness:
   a. Does the solution meet all the task requirements, including multi-layer monitoring?
2. Observability Solution:
   a. Is the telemetry stack well-chosen and effectively implemented?
   b. Are all layers of the stack being monitored?
3. Alerting & Monitoring:
   a. Are alerts meaningful and actionable?
   b. Are the dashboards intuitive and informative?
4. Documentation Quality:
   a. Is the documentation comprehensive and easy to follow?
   b. Can someone reproduce the environment and monitoring setup?

Create a Report:

Commit your work into a private GitHub repository and share it with @ahmedshafik Before diving in, please share your estimated time of completion for these tasks with Laura.

How is this task related to the position:

At Veriff we build and maintain scalable, high-performance, and highly available systems on a daily basis. It is essential for all engineers who join the team to have a solid understanding of the fundamental parts of any IT system. The team is expected to be able to create, monitor, and maintain our infrastructure while understanding all the surrounding products. This ensures that our systems and infrastructure are carefully monitored and dependable, as we hold the responsibility for the system's backbone.