**Cloud Automation Test Engineer**

**Recruitment Quiz**

Thank you very muchfor filling out this quiz to the best of your competencies. The purpose of this quiz is to evaluate your areas of expertise.

If a question is unclear, feel free to rephrase it or lay new assumptions, but don’t forget to document them.

Please provide your writeup in *.pdf* format, along with the source code and other relevant files packaged in a single zip archive named like *firstname\_lastname\_CATE.zip*. If the package is larger than 10Mb please provide a link to download it.

**Good luck and thank you again for your time!**

# v1.1

# Question 1

Consider the log file engine.log presented in folder Question1. This file contains the logs of the unix process nxengine starting and running.

In the log:

• starting nxengine indicates that the process is starting.

• nxengine is running indicates that the process has started to run.

Write a script (feel free to use any language you are comfortable with; bonus point if it’s not Java) to list how long nxengine takes to start running. Output the result in the following format:

Start Cycle: 1 Duration: HH:MM:SS  
Start Cycle: 2 Duration: HH:MM:SS  
...  
Start Cycle: n Duration: HH:MM:SS

# Question 2

What do you think are the most important characteristics of an automated test suite? Why? Please limit your answer to two paragraphs.

# Question 3

In your experience, what makes a product better or more successful when adopting a DevOps philosophy and methodology? How do you approach such adoption? Where do you start? Please provide an example from your professional experience. Please limit your answer to two paragraphs.

# Question 4

Consider the following tables

A picture containing text

Description automatically generated

1. What does the following query return?

SELECT department, avg(salary)

FROM Employees

GROUP BY department

HAVING department IN (

SELECT department

FROM Employees

GROUP BY department

HAVING count(\*) IN (

SELECT max(myCount)

FROM (

SELECT count(\*) AS myCount

FROM Employees

GROUP BY department)));

1. Write an SQL query that returns the number of employees per building.

**Question 5 (Microservices)**

Diagram

Description automatically generated

**Small description of the system:**

The end goal of this system is to digest data reliably from “Event feeders” and store them in a general database. It also provides a way for services to listen to events in real time via the event bus (something like Kafka).

One of the features shown in this diagram is a “Dashboards” application service. It reads data from the database and provides a web UI with metrics and graphs.

This feature consumes APIs provided by different teams. These APIs are a collection of core APIs like authentication and configuration provider services.

**Assignment:**

Given the architectural diagram and description above, describe what are the things you will be doing to test the whole system to ensure quality and reliability. Feel free to discuss the strategies for the whole system and for the services that your team will be working on (for example, Data Processing and Application services components). And if you have experience with testing distributed systems at scale, make sure to mention challenges and lessons learned!

This is an open-ended question, so make sure to structure your answer and focus on the aspects that in your opinion matter the most.

**Question 6 (Kubernetes)**

As described in Question 6, your “Dashboards” microservices are running in a Kubernetes cluster (shown as the blue square in the ‘Application Services’ layer). Unfortunately, it is not working as expected and nothing is displayed in the UI.

1. What are the first steps of your investigation? What could be some likely problems?
2. Following your investigation, you notice that a pod is in “CrashLoopBackOff” state. What do you do?
3. By checking the application logs of “Dashboards” service, you see the following error: “Can’t connect to auth-svc:4444”. What do you do?

**Question 7 - Bonus**

Create a small ansible project that writes the contents of a variable (command line environment variable or using ansible’s way of declaring variables) into an output file /tmp/var\_output.ansible located on the local machine. The requirements are the following:

* The main playbook must use a role where the actual write operation is performed
* When the variable is written in the output file and only if its value changed, the service “myservice.service” should be restarted (be aware of the permissions for such operations)
* The project must be maintainable and structured, meaning that other roles or other variables could be added in the future.

# Question 8 (Java)- Bonus

Consider the code presented in folder Question5 of the attached Java resources.

The program is a simple expression evaluator able to compute integer

expressions based on four operators (add, subtract, multiply and divide). The

logic is implemented in class Expression, which offers the following features:

* Evaluating the expression to retrieve its Integer value
* Pretty printing the expression
* Exporting the expression to XML.

The goal of this exercise is to refactor the Expression class (and optionally the

main program) to improve the code and to write unit tests to validate the program is functioning in the proper way.

You can change the code in any way you want, keeping in mind that:

* In the main program, we still want to create expression programmatically (no need to implement a parser)
* The 3 features described above must behave like in the original code, specifically:
  + No matter how the API for constructing the expression is implemented, it must still take care of invalid input.
  + The output of pretty-printing the expression must be the same as now.
  + The output of XML serialization must be the same as now.

You can add classes and change the API as you wish, as long as the previous points are not violated.

What we’re looking for is **clean**, **solid**, easy to understand, and **correct** code. Also

think about which design pattern can help you make the code more readable and more testable. In essence this task will require some level of **refactoring** in order to be successfully carried out.