HANSEN
POWER THE NEXT.™

DevOps New Hire Assessment

Contents

Introduction

This document outlines a series of exercises to be completed as a part of the new hire evaluation process for a role related to DevOps. Key prerequisite skills are:

- AWS – Associate DevOps or Architecture level
- Terraform – administrative level
- Coding and scripting

In this exercise, the script technology used is Python. If the candidate is not familiar with Python, the candidate should still be able to follow and modify the code leveraging their wider coding skills in other languages and technologies.

The candidate is expected to use a personal AWS account where applicable. There should not be any costs incurred as all tasks should occur well within free tier allowances.

You should anticipate spending about 6 hours completing all tasks.

Evaluation Exercise

A developer is trying to deploy the following solution in isolation so that, once written and unit tested, it can be included in the base architecture and subsequently deployed in every AWS EKS cluster. Here is the basic solution schematic:

1. In AWS Managed Prometheus (AMP), an alert event occurs. This is **not** a part of the solution the developer is working on as AMP has already been implemented in the base architecture.
2. An SNS topic is configured to which the Alert Event from AMP is sent. At minimum, the SNS topic has the lambda function configured as a subscriber.
3. From the SNS topic, the lambda function is invoked and the SNS message is passed in as input.
4. The lambda function processes the input, extracts out key alert information, and pushes this extracted information into DynamoDB

Task 1:

The developer has written the lambda function (see ReceivePrometheusSNSNotifications.py file in the provided zip archive) below and has provided the test input, but it is not working. The task is to:

1) Troubleshoot the function and make it operational

2) Show the **4** items in DynamoDB where attribute cluster_name is “test-cluster”. As can be seen from the test input below (see LambdaTestInput.txt file in the provided zip archive), 3 of the alerts are in the “Firing” state and 1 alert is in the “Resolved” state.
3) Show the CloudWatch Logs output.

Hint, the function should be configured with an SNS Topic Notification test template for ease of testing.

Specifications:

DynamoDB table schema:

- table-name: EKS_cluster_monitoring
- partition key: cluster_name = string
- sort_key: alert_name = string

Lambda Runtime: Python 3.9

Lambda function source code: ReceivePrometheusSNSNotifications.py (file in zip archive)

Test Input to Lambda function: LambdaTestInput.txt (file in zip archive)

Deliverables:

- Working Python script

Task 2:

Implement Terraform scripting to deploy this Proof of Concept. It should include all elements of Task1: SNS, Lamdba, DynamoDB. Specifically, the scripting must cover:

1) Setting up the SNS topic
2) Installing the working Lambda Function from above
3) Subscribe the Lambda function into the SNS topic
4) Deploying the DynamoDB table outlined in the previous task
5) All IAM permissions as required including CloudWatch logs

The solution can them be tested by manually publishing the Test Message, outlined in the specifications below, into the SNS topic.

Upon completion, provide to us the Terraform scripting ready for independent testing. After execution, the alert data from the message should be inserted into DynamoDB and CloudWatch logs should contain the log output including the print statements.

Specifications:

SNS details

Topic Name: `test-cluster-PrometheusAlerts`

Display Name: `AWS Prometheus Alert for test-cluster`

Test Message to pass into the SNS topic for testing (same message included in lambda test input above):

```
Alerts Firing:
Labels:
 - alertname = NodeCountNeverZero
 - app_kubernetes_io_component = metrics
 - app_kubernetes_io_instance = prometheus-for-amp
Annotations:
 - description = Node count over period has never been zero
 - summary = Nodes always running (instance 10.0.2.205:8080)
 - alertname = NodeCountNeverZero
Labels:
 - alertname = PrometheusNotConnectedToAlertmanager
 - instance = localhost:9090
 - job = prometheus
 - severity = critical
Annotations:
 - description = Prometheus cannot connect the alertmanager
Labels:
 - alertname = NodeCPULow
 - severity = warning
Annotations:
 - description = Little CPU activity detected on nodes
  VALUE = 0.823870370369971
Alerts Resolved:
Labels:
 - alertname = IngressByteRateZero
 - severity = warning
Annotations:
 - description = No ingress activity detected on product ingresses
 - summary = No ingress activity (instance )
```

Deliverables:

- Working Terraform script ready for testing

Task 3:

Provide installation and guidance documentation for Task 2. This should be suitable for a client or someone with suitable Terraform and AWS experience, but no direct familiarity with the product. You may format this as you think best and use any diagrams you think helps improve clarity. The person should be able to perform the installation with no assistance and no guess-work. Make other assumptions as you judge best.

Deliverables:

- Installation and guidance documentation for Task 2

Task 4:

This task is about demonstrating thought leadership. At Hansen, it is important demonstrate leadership qualities and to seek better, more effective solutions always. We are not artificially bound by technology constraints or corporate technology policies (beyond security) and our clients look to us for guidance.

The scenario is that we are a company with an older product implemented in technologies only suitable for on-premises deployment. However, we desire a much more modern approach in almost every regard: product technology, architecture, development and release processes, toolsets. You are our newly hired expert on DevOps. We know that DevOps is central to what we are looking for – even driving key architectural decisions to better enable the processes that DevOps promotes. However, none of us have sufficient and hands-on experience

In approximately 1000 words, plus any supporting images, outline a vision of what can be achieved and the key benefits. How can we achieve higher quality, higher delivery velocity and any other benefits? Make any assumptions of the current situation you choose. The audience are middle managers that need an overview understanding of the above before they can support the transition and request the investment.

Guidance:

1. this task is about a migration to DevOps from some unspecified legacy approach to software development, not about AWS technologies and their solutions. However, you are welcome to refer to specific technologies from any platform (e.g., open source) for examples
2. Focus on tangible benefits. For example, implementing microservices allows for a range of benefits including better support for CI/CD through being able to deliver at the service level, rather than the entire product.

Deliverables:

- Outline document of the likely value realized by implementing DevOps