SRE Challenge

Introduction

Please answer questions with the following instructions:

- Questions 1 Must be answered all
- Questions 2: Must be answered all
- Questions 3-4: Choose one out of two questions
- Questions 5-6-7: These are bonus question feel free to answer one or more or skip

If you decide to answer all the questions, it is always good.

Please try to give a really detailed answer as much as you can including the names oftools, approaches, and assumptions.

Thank you for your time and Good Luck!

The answers must be submitted as a doc file with any executable scripts or as GitHub repo withREADME.md per task.

Story (Concise)

- Azure Subscription
- Complex microservices architecture with 60+ backend services
- RDBMS Database
- Redis Cache
- Service Bus
- Backend - .NET
- Frontend - JS
- Docker + Kubernetes

Question 1 (Level 1):

Please write automation for infrastructure creation in Azure Cloud for:

1. Azure Resource Group
    - name – sre-challenge-flashenpost
    - tag – department: SRE
2. Azure Storage account
    - name – srechallengeforflaschenpost
    - resource-group - sre-challenge-flashenpost
    - SKU – Standard_LRS
    - Access tier HOT
    - tag – department: SRE
3. Storage Account Container
    - name – sre
    - type - private
    - storage account name – srechallengeforflaschenpost
4. Please add outputs to expose
    - Storage account ID
    - Storage account Primary access key
    - Storage account Primary connection string
    - Container ID

Assume:

• It is enough to save the output of the terraform plan , without actual creation of the infrastructure

• You use Terrafrom v1+

• You commit state file locally

• You use Azure West Europe Region

Note: You might need to fix any problems that occur while trying to plan your terraform code

---

Question 2(Level 2):

Your application team has developed a shop backend API, which runs on port 80.

The application team has asked you to deploy the created container image, which has to be deployed on a Kubernetes cluster.

Your task:

1. Write a Kubernetes Deployment definition according to the expectations listed below:

   • The Deployment is configured to deploy a minimum of three Pod replicas; Every Pod runs a single container, strm/helloworld-http:latest

   • The container exposes port 80;

   • The container must have configured a LivenessProbe, which makes TCP requests to the port 80;

   • Additional configurations on the LivenessProbe:

   - Set initial probe delay to 10 seconds;

• The container must have configured a ReadinessProbe, which makes HTTP requests to the “/” endpoint on port 80;

   • Additional configurations on the ReadinessProbe:

   - Set initial probe delay to 10 seconds;
   - Set check period to 1 second;
   - Set max probe failures to 2;

2. Set up a CronJob that outputs "Hello World" every thirty minutes.

For this test assume that:

The Deployment && Cronjob will be created in the default namespace (it is not expected to define its own namespace);

Your solution will be applied using HELM or kubectl cli

The file you are editing should be written as a valid Helm chart or YAML files

Use Deployment object from apiGroup apps/v1

The Deployment will run on Kubernetes v1.20.

Note: You might need to debug any problems that occur while trying to have this serve traffic. Document any fixes required in either a separate text file or alongside your code.

Question 6(Level 3):

In this challenge, write a program to analyze a log file and summarize the results. Given a text file of an http requests log, list the number of requests from each host. Output should be directed to a file as described in the Program Description below.

The format of the log file, a text file with a .log extension, follows. Each line contains a single log record with the following columns (in order):

1. The hostname of the host making the request.

2. This column's values are missing and were replaced by a hyphen.

3. A timestamp enclosed in square brackets following the format [DD/mmm/YYYY:HH:MM:SS -0400], where DD is the day of the month, mmm is the name of the month, YYYY is the year, HH:MM:SS is the time in 24-hour format, and -0400 is the time zone.

4. The request, enclosed in quotes (e.g.,"GET /media/catalog/product/cache/d2382938d1fed2fa0e610de14ff83091/1/0/103577-24-0_16596.jpg HTTP/1.0").

5. The HTTP response code.

6. The total number of bytes sent in the response

Log file sample- https://fpsrechallenge.blob.core.windows.net/sre/fp-sre-challenge.log

We love the code being readable, having comments and a good structure ;)

Question 7(Level 3):

In this challenge, write an HTTP GET method to retrieve information from a films DB

Given a string substr, the function getMovieTitles must perform the following tasks:

1. Query https://jsonmock.hackerrank.com/api/movies/search/?Title=substr (replace substr, You can check (https://jsonmock.hackerrank.com/api/movies/search/?Title=Superman)).
2. Initialise the titles array to store total string elements. Store the Title from each record returned in the data field to the titles array.
3. Sort titles in ascending order and return it as the answer.

The query response from the website is a JSON response with the following five fields:

- page : The current page
- per_page : The maximum number of results per page
- total : The total number of records in the search result
- total_pages : The total number of pages which must be queried to get all the results
- data : Data: An array of JSON objects containing movie information where the Title field denotes the title of the movie. In order to get all results, you may have to make multiple page requests. To request a page by number, your query should read https://jsonmock.hackerrank.com/api/movies/search/?Title=substr&page=pageNumber, replacing substr and pageNumber

We love the code being readable, having comments and a good structure ;)

Question 5(Bonus):

What triggers for scalability would you use for:

- Backend servers
- Database performance
- Service Bus performance (RabbitMQ | Azure Service Bus)
- Docker + Kubernetes

Please use a detailed textual description per above mentioned case/service

---

Question 6(Bonus):

Draw build & deploy process including CI\CT\CQ\CD for the Backend application (.Net \ Scala \ Python) from the developer's computer till the production release into Azure Kubernetes Cluster (AKS)

Please try to be as detailed as you can

Question 7(Bonus):

Please describe how you aggregate logs in Microservices architecture system from:

- Frontend
- Backend
- DBs
- Message Bus

Please describe what types of logs would you store and what tools you would use for accessing them

You can use a diagram or textual description
