DevOps Test

Introduction

The following is a list of challenges meant to evaluate your knowledge with various technologies. You can complete as many of these challenges as you want, it is not a requirement to complete all of these. For challenges that require a diagram please submit the diagram as an image file with the name and number of the challenge. For all the rest, you can collect your answers in a text document of your format and choice, anything from Word to markdown is accepted, but ideally format your answers in a manner that is presentable and easy to follow.

1.0.0 - Docker

1.1.0 - Create a Dockerfile that will build and generate a basic node express application.

- 1.1.1 - What docker command would you use to build the image for this dockerfile?
- 1.1.2 - What docker command would you use to run this image in interactive mode and allow me to view this express site from my browser ?
- 1.1.4 - What is the url I would need to input in my browser to view the above express site?
- 1.1.5 - What docker command would you use to run the this image in detached mode and allow me to view the express site from my browser?
- 1.1.6 - What docker command would you use to view the logs of the above container running in detached mode?
- 1.1.7 - What docker command would you use to run this image in interactive mode while allowing me to have access to the source code of this repository so that I can edit them in my preferred code editor?

1.2.0 - Create a docker compose file that will set up a fully functioning wordpress site.

- 1.2.1 - What docker command would I use to set up this docker compose configuration in interactive mode?
- 1.2.2 - What docker command would I use to set up this docker compose configuration in detached mode?
- 1.2.2 - What docker command would I use to view and follow the logs generate by this setup in detached mode?
- 1.2.4 - Once I have set up this wordpress site, if I destroy the container and recreate it, will I still be able to use the setup from my previous run? If not, what changes would you do to the docker compose file in order to make this possible?
- 1.2.3 - Can you make changes to this docker compose file in order to make it possible for developers to be able to make code updates to the wordpress site on their local code editor?

1.3.0 - The following package.json file has requirments on the version of node and npm needed to install application dependencies

```
{
  "name": "test-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "engines": {
    "npm": ">=8.0.0 <9.0.0",
    "node": ">=16.0.0 <17.0.0"
  },
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.18.2"
  }
}
```

- 1.3.1 - The package.json file state you need node version 16 to install the dependencies listed. You host operating system doesn't have this version installed or you have a version that doesn't match with the one listed. Write a docker command that would allow you to install the required dependencies with the required version of node on your host operating system (i.e. the package dependencies need to be persisted on the project folder even after the docker command has executed)

2.0.0 - Ansible

2.1.0 - You currently have the following servers running on a network. The following is a list of their ip address and the services that are currently running

- 10.0.0.20
  - MySQL
  - Host OS Ubuntu 18 LTS
- 10.0.0.21
  - MySQL
  - Host OS Ubuntu 18 LTS
- 10.0.0.30
  - nginx load balancer
  - Host OS Centos 8
- 10.0.0.40
  - backend api

  - Host OS Ubuntu 18 LTS
- 10.0.0.41
  - backend api
  - Host OS Ubuntu 18 LTS
- 10.0.0.50
  - frontend web application
  - Host OS Ubuntu 18 LTS
- 10.0.0.51
  - frontend web application
  - Host OS Ubuntu 18 LTS
- 2.1.1 - Write an ansible inventory file that will allow you to run ansible playbooks and commands.
- 2.1.2 - You are currently facing issues with disk space on your MySQL server getting full and would like to generate a daily report so that you are aware of when you are approaching disk capacity. Using the ansible inventory you defined, write an ad-hoc ansible command so that you can get this information
- 2.1.3 - Your backend api servers are running the service backend-api as a systemd process. This process is facing issues and you think a restart might help solve it. Write an ad-hoc ansible command that will allow you to restart this service.
- 2.1.4 - Your frontend web application servers currently have some log files located in /var/log/web/ that are taking up a lot of disk space. Log rotate has been configured but for some reason it has not been deleting old logs as intended, maybe due to a configuration issue. Assuming all log files marked for deletion have been compressed to zip files, write an ad hoc ansible script to delete all the old log files.
- 2.1.5 - You want to generate a daily report to check if any of the servers have recently underwent a restart. What ansible ad-hoc command would you use to check this?

2.2.0 - You have been tasked with setting up an ubuntu server to run a nodejs application as a service. Use the following article from digital ocean to create an ansible playbook so that you can set up this service https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-20-04. Since this task will require using multiple files and folders, create a private git repository and provide us with access to this repository.

- 2.2.1 - You made some changes part 4 of the article and want to rerun the ansible playbook without having to rerun all the previous steps. Can you the playbook you created do this, (if not what changes would you do to make this possible) and what command would you execute to only run this part?
- 2.2.2 - Your playbook is running correctly but for some reason the service is not accessible outside of the server. Can you add some tasks to the

playbook that would test if the reverse proxy is serving the application correctly?
- 2.2.3 - When running the playbook you want to be able to change the port that the application is running on by passing a parameter to the ansible playbook command, while also making sure that if the port changes the service will reflect this change. Can your playbook currently do this, and if it can, what is the command you would use to change the port?

3.0.0 - Linux

3.1.0 - You are currently experiencing issues with the services running on your Ubuntu server. Server performance is degrading and you need to investigate why. The services running on this server are nginx, php-fpm and some docker containers running a distributed micro service mesh with node, python and go runtimes.

- 3.1.1 - what method would you use to connect to server in order to troubleshoot the issue at the source?
- 3.1.2 - given that you still don't know what the issue might be, list a number of commands you would execute to better understand what is wrong with this environment, stating what insights this command would provide to you in helping you pinpoint the issue.
- 3.1.3 - you suspect the issue might be from the nginx service. How would you figure out where the error log is located and how will you inspect this log?
- 3.1.4 - you suspect the issue might be from the php-fpm. How would you figure out where the error log is located and how will you inspect this log?
- 3.1.5 - you suspect the issue might be from a docker container. Assuming the docker container is writing logs to the default log driver, how would you inspect the logs generated by this container?
- 3.1.5 - you suspect the issue might be from a docker container. Assuming the docker container is writing logs using the syslog log driver, how would you inspect the logs generated by this container?
- 3.1.6 - you suspect the issue might lie somewhere else. Which logs would you investigate to try and find out what the issue is?

3.2.0 - You have an api endpoint that will provide you with the following json file that contains information servers that you are managing.


```
{
  "data": [
    {
      "id": 1,
      "hostname": "host1",
      "ip": "10.0.0.1",
      "username": "admin1",
```

```
      "status": "up"
    },
    {
      "id": 2,
      "hostname": "host2",
      "ip": "10.0.0.12",
      "username": "admin1",
      "status": "down"
    },
    {
      "id": 3,
      "hostname": "host3",
      "ip": "10.0.0.24",
      "username": "admin1",
      "status": "down"
    },
    {
      "id": 4,
      "hostname": "host4",
      "ip": "10.0.0.19",
      "username": "admin1",
      "status": "up"
    }
  ]
}
```


- 3.2.1 - What command would you use to host a simple http server so that you can fetch the above json file using `curl`?
- 3.2.2 - Depending on if you managed to host this file on an http server, can you write a bash script that will convert this json file into a csv file?
- 3.2.3 - Write a bash script that will give me a list of all the server that currently have a status of `down`
- 3.2.4 - Write a bash script that will convert the above json file into the following format:

```
Host host1
  User admin1
  Hostname 10.0.0.1

Host host2
  User admin1
  Hostname 10.0.0.12

Host host3
  User admin1
  Hostname 10.0.0.24
```

```
Host host4
  User admin1
  Hostname 10.0.0.19
```

4.0.0 Cloud Infrastructure

4.1.0 - You are planning of setting up a scalable, highly available service using cloud servers on a platform of your choice.

- 4.1.1 - Create a diagram to illustrate which services you would set up to make this possible
- 4.1.2 - Write a short paragraph to describe the diagram and explain how this addresses common issues with failure recovery and scalability.

5.0.0 Terraform

5.1.0 - You have been tasked with creating a set of terraform configurations to set up some services on AWS

- 5.1.1 - Write a terraform configuration that opens ports 22 and 80 to the internet. Name the security group "WebServer-SG".
- 5.1.2 - Given the above Security group created in 5.1.1, write a Terraform code to launch an EC2 instance in AWS using this Security group. The EC2 instance should be a t2.micro instance running the Amazon Linux 2 AMI.
- 5.1.3 - You need to create an S3 bucket in AWS. Write a terraform configuration to create an S3 bucket named "terraform-test-bucket" in the AWS eu-central-1 region.
- 5.1.4 - Write a terraform configuration to create a VPC with two subnets. The VPC should be named "terraform-test-vpc", and the two subnets should be named "terraform-test-subnet-a" and "terraform-test-subnet-b". Both subnets should be created in the eu-central-1 region. The IP address range for the VPC should be 10.0.0.0/16. Subnet A should have an IP address range of 10.0.1.0/24 and Subnet B should have an IP address range of 10.0.2.0/24.
- 5.1.5 - Write a terraform configuration to create an Auto Scaling Group with launch configuration defined in task 5.1.2. The Auto Scaling Group should initially launch one EC2 instance and a maximum of two EC2 instances. The Auto Scaling Group should be launched in the VPC created in task 5.1.4 and the subnet should be selected randomly.
- 5.1.6 - Write a terraform configuration to create a Route 53 Record Set that sets the DNS name "terraform-test.com" to the public IP address of the EC2 instance launched in task 5.1.2.