Job Interviews

Technical Tasks

1. Write a dockerfile that packages a python script and runs it as a command
2. Write a docker-compose.yml file that starts wordpress application (wordpress is a blogging application with a MySQL backend)
3. Write an ansible role that installs 3 packages on the servers, checks if the servers are CentOS and if they do it will create an empty file under /tmp/this_is_centos
4. Write an ansible inventory file and show the use of group vars.
5. Write a jenkins pipeline that does the following:
    a. Checks out this repository: https://www.github.com/avielb/hw-14
    b. Run the playbook site.yml.
6. Write a python script that will run forever and every 5 seconds it will read a url from a file called url.txt and check if it is up or not. If it is not up it will print "<url> is dead" and close the program
7. Describe the steps to allow access from everywhere on port 8080 to your EC2 server.
8. Explain about the AWS CLI meaning and describe how to configure more than one IAM profile
9. Assuming I have a deployment in kubernetes of NGINX containers, what will be the commands to scale it from 1 to 3 pods
10. What will be the commands to deploy out-of-the-box redis with master and 3 replicas using helm.

Answers

1.

```
FROM python:3
ADD ex1.py /
CMD [“python”,”ex1.py”]
```

2.

```
version: '3.3'

services:
   db:
     image: mysql:5.7
     volumes:
       - db_data:/var/lib/mysql
     restart: always
     environment:
       MYSQL_ROOT_PASSWORD: somewordpress
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
       MYSQL_PASSWORD: wordpress

   wordpress:
     depends_on:
       - db
     image: wordpress:latest
     ports:
       - "8000:80"
     restart: always
     environment:
       WORDPRESS_DB_HOST: db:3306
       WORDPRESS_DB_USER: wordpress
       WORDPRESS_DB_PASSWORD: wordpress
       WORDPRESS_DB_NAME: wordpress
volumes:
    db_data: {}
```

3.

```
Mkdir -p roles/common/tasks/main.yml
---
- name: install
  package:
    name: {{ item }}
    state: present
  Loop:
    -  zip
    -  vim
    -  net-tools

- name: check if centos
  File:
    Path: /tmp/this_is_centos
    State: present
  when: ansible_distribution == 'CentOS'
```

4.

```
[servers]
Localhost
[servers:vars]
a=”b”
```

5.

```
Node{
  stages(){
    stage(“checkout”){
      Git ‘https://github.com/avielb/hw-14’
    }
    stage(“Run playbook”){
      Sh ‘ansible-playbook site.yml’
    }
  }
}
```

6.

```
import requests
from time import sleep
while True:
    r = open("url.txt")
    url = r.readline()
    i = requests.get(url.rstrip())
    if i.status_code != 200:
        print(url + " is dead")
    sleep(5)
    r.close()
```

7. Create a security group, define an inbound access rule on port 8080 and allow access from ‘Anywhere’

8. AWS CLI meaning is a tool that enables command line interface against AWS API. AWS CLI can be very helpful when we want to automate AWS commands and simple processes.

In order to configure multiple IAM profiles we can do it by adding the following

```
[default]
aws_access_key_id = AKIA*************
aws_secret_access_key =***************

[prod]
aws_access_key_id = AKIA***************
aws_secret_access_key = ****************

[dev]
aws_access_key_id = AKIA**************
aws_secret_access_key = ******************
```

9.

```
kubectl scale deployment/nginx --replicas 3
```

10.

```
helm install --name my-release --set cluster.slaveCount=3 stable/redis
```