Assignment

Assumption:

"rbcapp1" is a critical application that needs to be monitored and its status should be recorded into Elasticsearch. This Application depends on 3 services: httpd, rabbitMQ and postgreSQL .

If any of these services are down, "rbcapp1" state will be considered "DOWN" otherwise it is "UP".

Notes:

· Assume all these services runs under Linux machines.

· Include a README.md file to describe the answers and mention any special instructions if required to run the scripts

==========

TEST1: Assume all services are running on the same server as Linux services.

a) Write a Python script that monitors these services and creates a JSON object with application_name, application_status and host_name.

Sample JSON Payload

```
{
  "service_name":"httpd",
  "service_status":"UP",
  "host_name":"host1"
}
```

Write this JSON object to a file named {serviceName}-status-{@timestamp}.json

b) Write a simple Python REST webservice that:

Accepts the above created JSON file and writes it to Elasticsearch

Provide a second endpoint where the data can be retrieved, i.e

POST /add to Insert payload into Elasticsearch

GET /healthcheck --> Return the all Application status (“UP” or “DOWN”)

GET /healthcheck/{serviceName} --> Return the specific Application status (“UP” or “DOWN”)

Sample calls

https://myservice.example.com/add

https://myservice.example.com/healthcheck

https://myservice.example.com/healthcheck/httpd

==========
==========
=

==========
==========
=

TEST2: Assume the 3 services are running on different servers as RHEL services: httpd on host1, rabbitMQ on host2, postgreSQL on host3

a) Create an Ansible inventory file for the above hosts that meets the monitoring needs explained above

b) Write an Ansible playbook that will action based on a provided variable named "action":

"action=verify_install": verifies the services are installed on their allocated hosts and if not, the playbook should install it. (for the install, please pick just one service to illustrate)

"action=check-disk" : with this action it should check the disk space on all servers and report any disk usage > 80%. Send an alert email to a selected email address (Pick your own).

"action=check-status": with this action it should return the status of the application “rbcapp1” and a list of services that are down. (you can use the REST endpoint created in TEST1).

Below is a sample command to run the playbook

ansible-playbook assignment.yml -i inventory -e action=verify_install ---- This is for verify install as an example

==========
==========

==========
==========
=

TEST 3:

Enclosed CSV file(sales-data.csv) lists real estate sale prices.

Using Python, write a script to generate a new CSV file that only includes properties sold for less than the average price / foot.

========== End of Assignment
==========