The main target of this task is to check the candidate ability to learn and use frameworks and tools based Linux OS.

For this task, we will choose the famous ELK stack framework (Elastic Search + Kibana + Beats).

The following steps are required:

1. Create Linux VM based on Ubuntu 20.04 using HyperV, If the candidate is familiar with cloud service (GCP, AWS, digital ocean, ... etc) and use one of them it will be great, but it is not a necessary.
2. Install **ELK 8.2.0** (Elastic Search + Kibana), please don't use Docker for this step.
3. Check the Kibana interface and make sure it is working (usually on http://localhost:5601).
4. Install Nginx and add the needed configurations to run Nginx as reverse http proxy, to access Kibana on port 8888 (http://localhost:8888)
5. Create new role named "Safee" and new user named "test" and assign the role "Safee" to this user.
6. Install metircbeat agent on the same machine, and configure it to send system metrics data to local elastic then enable the nginx module (You **should** setup metrics dashboard to check the metrics data using Kibana).
7. Update the "Safee" role to get access to metrics data (read **ONLY**).
8. Install filebeat agent on the same machine, and configure it to send logs to local elastic then enable system and audit modules.
9. Update the "Safee" role to get access to logs data (read **ONLY**).
10. Install heartbeat agent on the VM and add monitor to our Kibana service on port 5601 and another monitor for elastic service on port 9200, check the Uptime interface to make sure your setup is ok.

All previous steps should be so easy, just following the installation guide step by step, the following steps shouldn't be harder too and they are required:

1. Create a dashboard that shows:
    a. Most requested sites throw our proxy (should be Kibana on port 8888 and default site on port 80)
    b. Heartbeat counts per monitor.
    c. Heartbeat Status percentage per monitor (success, failed)
    d. Nginx access and error log (full details).

2. Create a dashboard that shows:
    a. SSH connections to our VM that shows list for:
        i. Username
        ii. Source IP
        iii. Status (success , failed)

b. Pie chart for SSH connections status percentage (success, failed)

3. Give role “Safee” access on the two dashboards; ***do we need to do anything here?***

For alerting purposes we will use an external tool called elastaert2, the following tasks are required:

1. Run elastaert2 on the same VM using Docker-Compose.
2. Add an alert rule to send an email when the VM CPU average is higher than 50% for the last 5 min. (you **should** depend on mertbeats data from step 6).
3. Add an alert rule to send an email when any heartbeat monitor failed once in the last 5 mins.
4. Add an alert rule to send a message to telegram channel for every SSH Connection. The message should contain the connection status (success or failed)

Bounce: The candidate will deliver a wiki page on public Github repo, that provide *well documentation for all configuration and actions he used to fulfil all tasks.*

Hint*: to win the bounce the candidate can easily write down everything while he is working on the tasks.*